TU Clausthal

Framework for Efficient and Accurate Simulation of the Dynamics of Machine Tools

Dissertation
to be awarded the degree

Doctor of Engineering (Dr.-Ing.)

submitted by

Daniel Spescha

from Ilanz/Glion, Switzerland

approved by the
Faculty of Mathematics/Computer Science and Mechanical Engineering
Clausthal University of Technology

Date of oral examination
09.03.2018

Chairperson of the Board of Examiners
Prof. Dr.-Ing. Volker Wesling

Chief Reviewer
Prof. Dr. habil. Oliver Zirn

Reviewers
Prof. Dr.-Ing. Stefanie Retka
Prof. Dr.-Ing. Konrad Wegener
Prof. Dr.-Ing. Stefan Hartmann

Abstract

The aim of this dissertation is to formulate a comprehensive framework of methods, algorithms, and techniques for efficient and accurate simulation of the dynamics of machine tools, to implement these methods in a software package, and to give advise of how to use simulation analyses effectively during development of new machine tool products.

Relevant properties of machine tool structures are derived and a thorough literature study reveals that there is a lack of methods for efficient model order reduction, reduction error estimation, and modelling of moving interfaces which are compatible with reduced-order models. Furthermore, there are no complete frameworks for the simulation of moving and rotating axes with reduced-order models.

A new model order reduction method which is based on Krylov and modal subspace projection is presented. An in-depth analysis of the quality of approximation of them both shows that a combination of them leads to a beneficial combination of their properties. Krylov subspace based reduction allows matching the frequency response function at specific frequencies and modal subspace projection leads to accurately matching poles of the frequency response function. The frequency response functions of the resulting reduced models allow the estimation of an upper bound for the relative error. The result is an a-priori error estimation which depends on the number of Krylov iterations per expansion point, the expansion points, and the number of eigenvectors used for reduction only. The quality of the estimation is analysed using randomly generated systems as well as using a real-world example.

Modelling of moving interfaces in combination with reduced-order models is challenging because changing the location of action of an interface on a flexible body involves changing the finite element nodes involved. Using all potentially loaded nodes as independent inputs and outputs to a system is not feasible for accurate model order reduction methods, because every input and output to a system enhances the number of degrees of freedom for the reduced system. Therefore, a new method for the approximation of a density function along a path by means of trigonometric interpolation is presented. This method leads to a minimum number of independent interfaces which represent the harmonics of a Fourier series. After having reduced the models, the harmonics can be superposed in order to achieve the desired density function, e.g. a trapezoidal force at the location of a linear guide carriage.

Creation of system matrices for rotated bodies is addressed using a floating frame of reference formulation. Therefore, an application-oriented derivation of the required information for the assembly of system matrices of the bodies in any orientation in space is presented.

The methods above are embedded in a comprehensive software package that interfaces with ANSYS Mechanical for the creation of finite element models and with MATLAB and Simulink for modelling of the complete mechatronic system. Special attention is paid to an effective work-flow. Changes in any part of a model affect this part only. Neither a time-consuming computation nor a cumbersome manual task have to be repeated, if it is not logically required.

In order to speed up the process of designing or redesigning a machine tool structure, the Design to Specifications approach is introduced. This approach describes the derivation of requirements on the structure of a machine tool used for the fulfilment of the specifications on productivity and accuracy. Modelling with the presented methods is started as soon as a first design is finalised and for an efficient evaluation of the actual dynamic performance, a method based on weighted error transfer functions is presented.

For verification, a test bench is modelled and compared with measurements. Despite not having fitted any parameter for optimising the accordance between measurement and simulation, the results are accurate in qualitative and quantitative terms. The methods for derivation of requirements as well as the analysis of the dynamic performance are applied to the model of the test bench and are proved valid.

To summarise, this thesis gives a complete framework consisting of simulation methods, algorithms, software, and application techniques, leading to an exceptionally efficient work-flow for the simulation and design of machine tools.

Zusammenfassung

Das Ziel vorliegender Dissertation ist es, Methoden, Algorithmen und Anwendungstechniken zu entwickeln, welche eine effiziente und akkurate Simulation des dynamischen Verhaltens von Werkzeugmaschinen ermöglichen und effizient im Entwicklungsprozess für Werkzeugmaschinen eingebettet werden können.

Eine vertiefte Literaturrecherche deckt auf, dass es an Methoden für effiziente Modellreduktion mit Fehlerschätzung sowie für die Abbildung wandernder Schnittstellen fehlt.

Eine neue Modellreduktions-Methode, die auf Projektion in kombinierte Krylov und modale Unterräume basiert, wird präsentiert. Es wird gezeigt, dass eine Kombination dieser Methoden zu einer vorteilhaften Kombination derer Eigenschaften führt. Krylov Unterraum basierte Reduktion erlaubt den Abgleich des Frequenzgangs für spezifische Frequenzen. Modale Reduktion hingegen führt zu präzise übereinstimmenden Polstellen der Übertragungsfunktionen. Die Einfachheit der resultierenden Fehlerfrequenzgänge erlaubt die Schätzung der Obergrenze des relativen Fehlers im interessierenden Frequenzbereich. Das Ergebnis ist eine a priori Fehlerschätzung, welche nur von einer geringen Anzahl an Parametern der Reduktionsmethode abhängt. Die Qualität der Schätzung wird mittels zufällig generierten Systemen, wie auch anhand eines Beispiels aus der Praxis analysiert.

Die Modellierung von wandernden Koppelstellen in Kombination mit Modellordnungsreduktion ist herausfordernd, da eine Verschiebung des Kraftangriffs auf einem flexiblen Körper auch eine Änderung der beteiligten finite Elemente Knoten zur Folge hat. Alle potenziell involvierten Knotenfreiheitsgrade als unabhängige Ein- resp. Ausgänge zu verwenden ist in Kombination mit akkuraten Modellreduktions-Methoden nicht umsetzbar, da jeder Schnittstellenfreiheitsgrad zu einer Erhöhung der Ordnung des reduzierten Systems führt. Darum wird eine Methode präsentiert, die auf Approximation einer Dichtefunktion entlang eines Pfades mittels trigonometrischer Interpolation basiert. Diese Methode führt zu einer minimalen Anzahl an Schnittstellenfreiheitsgraden, welche den Harmonischen einer Fourier-Reihe entsprechen. Superposition dieser Harmonischen führt dann zur gewünschten Schnittstellenfreiheitsgrade für die ordnungsreduzierten Modelle.

Die Berechnung von Systemmatrizen für flexible Körper in beliebiger räumlicher Orientierung wird durch Aufspaltung der Körperverschiebungen in Deformationen im körperfesten Bezugssystem und Verschiebungen resp. Rotationen des Körpers im globalen Bezugssystem formuliert. Es wird eine anwendungsorientierte Herleitung der benötigten Informationen für die Bestimmung der Systemmatrizen für beliebige Posen des Körpers präsentiert.

Ein umfassendes Softwarepaket wird präsentiert, welches die erwähnten Methoden implementiert. Besonderes Augenmerk wird auf effiziente Arbeitsabläufe gelegt. Änderungen in einem beliebigen Teil des Modells wirken sich ausschließlich auf den betroffenen Teil aus. Weder zeitaufwändige Berechnungen noch umständliche manuelle Arbeit müssen wiederholt werden, wenn es nicht logisch erforderlich ist.

Um den Prozess der Neu- oder Weiterentwicklung einer Maschinenstruktur zu beschleunigen, wird der als Design to Specifications bezeichnete Ansatz eingeführt. Dieser Ansatz beschreibt die Herleitung von Anforderungen an die Struktur einer Werkzeugmaschine, die für die Einhaltung der Spezifikationen zu Produktivität und Genauigkeit erfüllt werden müssen. Mit der Modellierung anhand der präsentierten Methoden wird begonnen, sobald ein erster Entwurf der Konstruktion erstellt wurde. Für die effiziente Beurteilung der dynamischen Leistungsfähigkeit der Maschine wird eine Methode präsentiert, die auf gewichtete Fehlerübertragungsfunktionen basiert.

Zwecks der Verifikation werden Ergebnisse aus Simulation und Messungen eines Prüfstands verglichen. Auch ohne gezielte Parametermodifikation, d.h. unter Verwendung der Herstellerangaben für Koppelstellenparameter etc., kann sowohl qualitativ als auch quantitativ eine gute Übereinstimmung der Mess- und Simulationsergebnisse nachgewiesen werden. Auch die Methoden für die Herleitung von Anforderungen an die Struktur und für die Analyse der dynamischen Leistungsfähigkeit werden auf das Modell des Prüfstands angewandt und verifiziert.

Zusammenfassend präsentiert diese Dissertation ein umfassendes Rahmenwerk für einen überaus effizienten Arbeitsablauf bei der Simulation und Konstruktion von Werkzeugmaschinen.

Acknowledgements

Thanks to the support of many people, I've had the opportunity of realising this dissertation during my time at inspire AG in Zurich.

This thesis would not have been possible without the support of my supervisor Prof. Dr. Oliver Zirn, who trusted in me and coached me all the years for a successful finalisation of the thesis. Moreover, I could only realise the thesis with thanks to kind support of my advisor Prof. Dr. Konrad Wegener. I appreciate all the valuable discussions we have been having, but I am especially grateful for he having supported my PhD thesis and having supervised me without making any difference to his own PhD students. I also want to thank Prof. Dr. Stefanie Retka and Prof. Dr. Stefan Hartmann for reviewing my thesis.

Many thanks go also to my group leader Dr. Sascha Weikert for his trust in my work, his support over the last years, and for teaching me many topics in the field of machine tools with a profound knowledge.

Furthermore, I want to thank the people who inspired me to delve into a PhD study. This is first, Prof. Dr. Ruprecht Altenburger, who was my advisor during my master's studies. He introduced me into the matter of simulation, model reduction, and machine tools. Moreover, I want to thank Dr. Roman Süsstrunk, with whom I carried out several theses during the diploma studies. Our ambitious work during this time inspired me for more. Furthermore, I would not have started a PhD study without the encouragement and the kind advice of Dr. Sergio Bossoni.

I want to thank my colleagues for all the valuable discussions in our office, during collaboration in projects, as well as during sport and leisure activities, especially Natanael Lanz and Dr. Markus Steinlin for giving constructive feedback on topics around simulation, control theory, and software implementation, as well as Pablo Hernández-Becerro for proofreading the thesis. Furthermore, I appreciate the feedback on software functionalities and usability I got from my colleagues Dr. Michal Kuffa, Dr. Titus Haas, and Nino Ceresa. I've experienced a great time at inspire AG with many fellow PhD students, for which I am very grateful.

Finally, I would like to thank my beloved wife Anja for patiently supporting and encouraging me during the time of my PhD studies.

Daniel Spescha
September 2018

Meiner wundervollen Frau

A mes geniturs

Nomenclature

Acronyms

BIBO	Bounded-input, bounded-output stable
CMS	Component mode synthesis
DAE	Differential algebraic equation
DOF	Degree of freedom
FRF	Frequency response function
KMS	Krylov and modal subspace
MIMO	Multiple-input, multiple-output system
ODE	Ordinary differential equation
SISO	Single-input, single-output system
TCP	Tool centre point

Notations

$\boldsymbol{B}$	Bold capital letters denote matrices
$\boldsymbol{b}$	Bold lower-case letters denote vectors
$\boldsymbol{b}_i$	Element indexing of vector
$\boldsymbol{B}_{i,j}, \boldsymbol{B}_{i:}, \boldsymbol{B}_{:i}$	Element, row, and column index of a matrix
b, B	Regular letters denote scalars
$[\bar{\cdot}]$	Upper limit for a scalar value or global representation of a vector defined in a local reference system
$[\hat{\cdot}]$	Reduced system matrices
$[\tilde{\cdot}]$	System in modal coordinates
$[\cdot]_f$	Denotes components associated with deformation
$[\cdot]_r$	Denotes components associated with rotation
$[\cdot]_t$	Denotes components associated with translation
$[\cdot]^{(k)}$	Indexing of finite element with number $k \quad (k \in 1, 2, \ldots, N_e)$
$[\cdot]^{[i]}$	Indexing of finite element node with number $i \quad (i \in 1, 2, \ldots, N_n)$
$[\cdot]^{i,k}$	Indexing an entity k associated with body i
$[\cdot]_i^{(k)}$	Indexing of node i of the finite element element $k \quad (i \in \{I, J, K, L, M, N, O\}, \quad k \in 1, 2, \ldots, N_e)$
$[\cdot]_S$	Denotes an entity associated with the source interface of a link
$[\cdot]_T$	Denotes an entity associated with the target interface of a link

Mathematical Operations

$[\cdot] \succcurlyeq 0$	Positive semi-definite matrix

det	Determinant of a matrix
diag	Diagonal matrix
orth	Orthonormalisation of a matrix
range	Column range of a matrix, i.e. the image of a matrix
$\delta(t)$	Dirac pulse or Dirac delta function
$\mathscr{F}$	Fourier transform
$\mathscr{K}_m$	Krylov subspace of dimension m
$\mathscr{L}$	Laplace transform
$\tilde{\boldsymbol{r}}$	The tilde denotes a skew symmetric matrix associated with vector $\boldsymbol{r}$
$\bullet$	Scalar product
$\theta(t)$	Unit step or Heaviside step function
$\times$	Cross product
$u(i\omega) = \mathscr{F}\{u(t)\}$	Fourier representation of a function
$u(s) = \mathscr{L}\{u(t)\}$	Laplace representation of a function

Symbols

$(\varphi_u, \varphi_v, \varphi_w)$	Rotations in local coordinates
(ξ, η, ζ)	Local coordinates for a finite element surface
(u, v, w)	Local coordinates
(x, y, z)	Global coordinates
α	Mass proportional damping coefficient
$\hat{a}$	Maximum acceleration amplitude
$\boldsymbol{A_0}^{i,k}$	Rotation matrix for interface k of body i in the body reference system
$\boldsymbol{A} \in \mathbb{R}^{N \times N}$	System dynamics matrix
$\boldsymbol{A}^i$	Rotation matrix for body i
$a^*(t)$	Acceleration set-point
$A^{(k)}$	Surface of the k^{th} element
a_0	Constant term of a trigonometric interpolation
a_k	k^{th} cosine coefficient of a trigonometric interpolation
$Y(i\omega)$	Maximum set-point weighting without violating kinematic and dynamic constraints
β	Stiffness proportional damping coefficient
$\boldsymbol{B_G}^{i,k} \in \mathbb{R}^{N \times 6}$	Interface matrix associated with interface number k of body i expanded to the coordinates of the assembled system
$\boldsymbol{B_T} \in \mathbb{R}^{N \times p}$	Thermal input matrix

$\boldsymbol{b} \in \mathbb{R}^{N\times 1}$	Input vector (column vector)
$\boldsymbol{B} \in \mathbb{R}^{N\times p}$	Input matrix
$\boldsymbol{B}^{i,k} \in \mathbb{R}^{N\times 6}$	Interface matrix associated with interface number k of body i
$\boldsymbol{B}^{l}$	Interface matrix associated with link l
b_k	k^{th} sine coefficient of a trigonometric interpolation
$\boldsymbol{C_T} \in \mathbb{R}^{N\times N}$	Thermal capacity matrix
$\boldsymbol{c} \in \mathbb{R}^{1\times N}$	Output vector (row vector)
$\boldsymbol{C} \in \mathbb{R}^{q\times N}$	Output matrix
$\hat{d}_a$	Maximum dynamic tracking error for acceleration limited trajectories
$\hat{d}_j$	Maximum dynamic tracking error for jerk limited trajectories
$\boldsymbol{D} \in \mathbb{R}^{N\times N}$	Damping matrix
$\boldsymbol{D}^{l}$	Link damping matrix for link l
D_c	Position controller damping ratio
$\bar{\varepsilon}_p$	Upper limit for relative pole error
$\bar{\varepsilon}_r$	Upper limit for relative error
$\bar{\varepsilon}_z$	Upper limit for relative transmission zero error
$\hat{e}_c$	Maximum contour error
$\boldsymbol{e_d}$	Unit vector for direction d of a coordinate system
$\boldsymbol{E_M}(i\omega) \in \mathbb{C}^{q\times p}$	Relative error of $\boldsymbol{H_M}$
$\boldsymbol{E_r}(i\omega) \in \mathbb{C}^{q\times p}$	Relative error of $\hat{\boldsymbol{H}}$
$\boldsymbol{E_{RK}}(i\omega) \in \mathbb{C}^{q\times p}$	Relative error of $\boldsymbol{H_{RK}}$
$\boldsymbol{E} \in \mathbb{R}^{N\times N}$	Left-hand system matrix
ε_{p_k}	Relative error of the k^{th} pole frequency
ε_{z_k}	Relative error of the k^{th} zero frequency
$\hat{\boldsymbol{E}}$	Maximum in- and cross-talk error matrix
$\hat{\boldsymbol{f}}^{i,k}$	Interface load vector for interface k of body i
$\bar{\boldsymbol{F}}_{\boldsymbol{R}} \in \mathbb{R}^{6\times 6}$	Resulting load matrix of an the elementary load distributions of an interface
$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{0_d}}$	Constant nodal DOF vectors for the elementary force distribution in direction $d \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\}$
$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{ck_d}}$	k^{th} cosine nodal DOF vectors for the elementary force distribution in direction $d \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\}$
$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{sk_d}}$	k^{th} sine nodal DOF vectors for the elementary force distribution in direction $d \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\}$
$\boldsymbol{f} \in \mathbb{R}^{N}$	Force vector in state coordinates

$\boldsymbol{f}_d^{[i]} \in \mathbb{R}^3$	Nodal DOF vector for node i and direction $d \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\}$
$\boldsymbol{F_R} \in \mathbb{R}^{6\times6}$	Resulting load matrix of an interface
$\gamma^{(k)}$	Load for the k^{th} element
$\gamma_u^{(k)}$	Load for the k^{th} element
$\hat{g}(s)$	Trigonometric interpolation approximation of a weighting function $g(s)$
$\boldsymbol{G}(s) \in \mathbb{C}^{q\times p}$	Transfer matrix
$G(s) \in \mathbb{C}$	Transfer function
G_M	Plant transfer function
g_0	Constant term of a trigonometric interpolation
g_{ck}	k^{th} cosine harmonic of a trigonometric interpolation
g_{sk}	k^{th} sine harmonic of a trigonometric interpolation
$\hat{\boldsymbol{H}}(i\omega) \in \mathbb{C}^{q\times p}$	FRF matrix of a reduced system
$\boldsymbol{H_E}(i\omega) \in \mathbb{C}^{q\times p}$	Absolute error FRF matrix
$\boldsymbol{H_M}(i\omega) \in \mathbb{C}^{q\times p}$	FRF matrix of a modally reduced system
$\boldsymbol{H_R}(i\omega) \in \mathbb{C}^{q\times p}$	FRF matrix of the remainder system after modal reduction
$\boldsymbol{H_{RK}}(i\omega) \in \mathbb{C}^{q\times p}$	FRF matrix of the remainder system after modal reduction, reduced by Krylov reduction
$\boldsymbol{H}(i\omega) \in \mathbb{C}^{q\times p}$	FRF matrix of the original system
$H(i\omega) \in \mathbb{C}$	Frequency response function (FRF)
$\hat{j}$	Maximum jerk amplitude
$\boldsymbol{J}^i$	Inertia tensor of body i
$j^*(t)$	Jerk set-point
κ	Velocity loop gain
κ_{opt}	Velocity loop gain for optimal damping
$\boldsymbol{K_T} \in \mathbb{R}^{N\times N}$	Thermal conductivity matrix
$\boldsymbol{K} \in \mathbb{R}^{N\times N}$	Stiffness matrix
$\boldsymbol{K}^l$	Link stiffness matrix for link l
K_p	Velocity controller gain
K_v	Position controller gain
$\boldsymbol{m_e} \in \mathbb{Z}_e^n$	Number of Krylov iterations per expansion point
$\boldsymbol{M} \in \mathbb{R}^{N\times N}$	Mass matrix
m^i	Mass of body i
$m_e \in \mathbb{Z}$	Number of Krylov iterations for a single expansion point
$N \in \mathbb{Z}$	Order of an original system

$n \in \mathbb{Z}$	Order of a reduced system
$n_b \in \mathbb{Z}$	Number of bodies in a multi-body system
$N_e \in \mathbb{Z}$	Number of elements of a finite element model
$n_e \in \mathbb{Z}$	Number of expansion points
$n_K \in \mathbb{Z}$	Dimension of a Krylov subspace
$n_M \in \mathbb{Z}$	Dimension of a modal subspace
$N_n \in \mathbb{Z}$	Number of nodes of a finite element model
$n_z \in \mathbb{Z}$	Number of transmission zeros
$\boldsymbol{\phi_d} \in \mathbb{R}^N$	Rigid body motion shape for direction $d \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\}$
$\boldsymbol{\Phi_M} \in \mathbb{R}^{N \times n_M}$	Modal matrix containing the first n_M eigenvectors
$\boldsymbol{\Phi_R} \in \mathbb{R}^{N \times (N - n_M)}$	Modal matrix containing all but the first n_M eigenvectors
$\boldsymbol{\Phi_r} \in \mathbb{R}^{N \times 3}$	Modal matrix containing rotational rigid body modes
$\boldsymbol{\Phi_t} \in \mathbb{R}^{N \times 3}$	Modal matrix containing translational rigid body modes
$\boldsymbol{\Phi} \in \mathbb{R}^{N \times N}$	Modal matrix containing eigenvectors as columns
$\boldsymbol{p}(s) \in \mathbb{R}^3$	Moving path of a moving interface depending on parameter s
$\boldsymbol{p}_d^{[i]} \in \mathbb{R}^N$	Allocation vector for degree of freedom d of node i
$p \in \mathbb{Z}$	Number of inputs of a system
$q(x, y, z)$	Distributed load
$q \in \mathbb{Z}$	Number of outputs of a system
$\boldsymbol{R_F} \in \mathbb{R}^{6 \times 6}$	Orthonormalisation matrix for an interface
$\boldsymbol{r_0}$	Center of action of an interface
$\boldsymbol{r_C}^{(k)}$	Center of action of a load for the k^{th} element
$\boldsymbol{r_S}^{i}$	Center of mass of body i
$\boldsymbol{s_e}\,,\ \boldsymbol{\sigma_e} \in \mathbb{C}^{n_e}$	Expansion points
$\sigma \in \mathbb{C}$	Substitute for squared Laplace parameter ($\sigma = s^2$)
$s \in \mathbb{C}$	Laplace parameter or path parameter
$s_e\,,\ \sigma_e \in \mathbb{C}$	Single expansion point
$\boldsymbol{t}(x, y, z)$	Vector field
τ	Position controller lag
T_n	Velocity controller integrator time constant
$\mathcal{U}(a, b)$	Uniform distribution in the interval (a, b)
$\boldsymbol{u_T} \in \mathbb{R}^{q \times N}$	Heat flow input vector
$\boldsymbol{u} \in \mathbb{R}^p$	System input
$\hat{v}$	Maximum velocity amplitude

$\boldsymbol{V_K} \in \mathbb{R}^{N\times n_K}$	Basis matrix for a Krylov subspace
$\boldsymbol{V_R K} \in \mathbb{R}^{N\times n_{RK}}$	Basis matrix for a Krylov subspace for the remainder system after modal reduction
$\boldsymbol{V} \in \mathbb{R}^{N\times n}$	Basis matrix for model reduction by projection
$v^*(t)$	Velocity set-point
V_{ff}	Velocity feed-forward gain
$\omega \in \mathbb{R}$	Angular frequency
ω_c	Position controller bandwidth
ω_l	Maximum frequency of interest
$\Omega_l = [-\omega_l, \omega_l]$	Frequency range of interest
ω_m	Angular frequency of the first discarded mode for modal reduction
ω_k	Angular frequency of the k^{th} pole
ω_M	Antiresonance frequency at the encoder
ω_P	Resonance frequency
ω_T	Antiresonance frequency at the TCP
ω_{z_k}	Angular frequency of the k^{th} transmission zero
$\boldsymbol{x_T} \in \mathbb{R}^N$	Temperature state vector
$\boldsymbol{x} \in \mathbb{R}^N$	State vector
$\hat{y}$	Maximum position amplitude
$\boldsymbol{y} \in \mathbb{R}^q$	System output
$y^*(t)$	Position set-point
ζ_k	Damping coefficient of the k^{th} pole

Contents

1 Introduction

During the last decades, the machine tool industry has experienced an evolution to products with high dynamic performance and precision. This has been enabled through increasingly integrated systems and the use of innovative lightweight structures as well as modern drive and control systems. In development of these modern machine tools, simulation has continuously gained in importance. However, due to the increasing pressure to reduce the time-to-market of new products, an effective simulation work-flow regarding the modelling process, the computing time for simulations, the accuracy of the results, and the evaluation of simulation results is crucial.

1.1 Subject of the Thesis

This thesis aims to provide a framework for the efficient and accurate simulation of the dynamics of machine tools for application during the development process. The terms *accurate* and *efficient* simulation can be characterised as follows:

Accurate Simulation The accuracy of the simulation results depends on multiple factors like input data, scope of the model, accuracy of the model, and numerics of model generation and simulation. The goal is to find a modelling strategy which enables the accurate modelling of the dynamics of a machine tool including static and dynamic deformations, drives, and control. This incorporates the selection of an appropriate modelling method, the development of accurate and numerically stable algorithms, and suggestions on modelling of machine components.

Efficient Modelling To be efficient in the modelling process, a straightforward work-flow is very important. This is especially true, if not all modelling tasks are accomplished by the same person. For flexible multi-body modelling of a machine tool including control loop simulation, the finite element model is often prepared by one engineer, the multi-body simulation by a second, and the control simulation by a third one. In this case, loops from one engineering field back to another should not be necessary in order to debug the model. Debugging the model and performing plausibility tests should be possible at each stage during the modelling process without extensive work. It should be possible to test small modelling steps, in order to keep the tests simple and clear.

Moreover, changes in the model of a preceding modelling step, e.g. changes of the finite element model due to changes in design or concept, should be possible to be propagated to all further modelling steps with minimal effort for modification. Repeating the same modelling tasks should be avoided where possible.

Efficient Calculation Extensive calculations are required for both, model order reduction of finite element models and transient or frequency domain simulation. The model order reduction method is of special importance because it can be a major calculation task and the resulting models determine the calculation effort for a simulation. Therefore, an efficient model order reduction technique has to be used, which allows the derivation of models of low order from large-scale finite element models.

Efficient Evaluation Once the model is created, debugged, and an efficient simulation is possible, the user has to conduct analyses which allow to specify the quality of a specific design. Often, models are used for transient simulation of the machining of specific workpieces. This is, on the one hand, time consuming and, on the other hand, does not provide general information on the performance for arbitrary workpieces. Methods should be provided, which allow an efficient judgement on the performance of the machine tool for arbitrary workpieces. The focus in this thesis lies on the prediction of dynamic path deviations.

1.2 Properties of Machine Tools

Machine tools have special properties which are discussed here to help focusing the research on methods which are appropriate for this family of systems.

Small Strains The structure of machine tools is deformable; and it is crucial to model these structural deformations. However, the strains in structural components are usually small and the structural components can be considered linear-elastic. Moreover, the main damping sources for machine tools are in the coupling elements and drives. Material damping for single components can usually be modelled as linear viscous damping with sufficient accuracy.

Large Translations Moving linear axes undergo large translations. This leads to moving contact points between the individual axes, e.g. at the linear guides, ballscrews, or linear direct drives, and therefore, to structural changes of the machine tool. Today, machine tools are usually simulated in one pose, but the vibration behaviour of a machine tool including slowly changing operation points is rarely simulated.

Large Rotations Large rotation of the bodies is inherent in the motion of rotary axes. This leads, similarly to large translations of linear axes, to structural changes of the machine tool.

Distinct Interfaces Interfaces to bodies of machine tools arise at components as, e.g. bearings, linear guides, motors, gears, measurement systems, or tool-workpiece-interaction. The interfaces to bodies are distinct in terms of location and behaviour. The location of action can change, but is determined by the positions of the axes. The behaviour can be linear or non-linear and both forces and torques have to be applicable just like translations and rotations have to be evaluable.

Limited Excitation The excitation of machine tool structures is usually decaying with frequency. The structure is excited by the drives and process forces only. The excitation by the drives is limited due to the low-pass characteristics of the controllers and the excitation spectra of process forces usually decrease with increasing frequency. This allows to define a relevant frequency range of interest.

Mechatronic Systems Due to the integration of structural components, drives, and controllers, machine tools are mechatronic systems. Therefore, a simulation environment has to enable modelling of the properties of those mechanical, electromechanical, and electronic devices by means of a system simulation environment.

2 State of the Art in Machine Tool Dynamics Simulation

In this chapter, an overview of the state of the art in machine tool simulation is given. Methods for simulation of machine dynamics are characterised and evaluated, model order reduction techniques for the reduction of large-scale mechanical models derived from finite element simulation are presented, coupling of flexible bodies is discussed, and an overview over publications specific to machine tools is given.

2.1 Introduction

Altintas, Brecher, Weck, and Witt [2] illustrated the benefit of virtual machine tools in order to reduce the development time and thus enhancing the competitive ability of machine tool manufacturers. They stated that time- and cost-intensive manufacturing, testing, and optimisation of physical prototypes can no longer be afforded and can, at least partly, be replaced by virtual prototyping and verification.

Kadir, Xu, and Hämmerle [53] provided an overview of different approaches for modelling virtual machine tools, including mathematical modelling, virtual reality, and hardware interaction. They compared the application domains and capabilities of the methods, and credited the most comprehensive applicability to the mathematical modelling of the machine tool, what comprises computer aided design (CAD), computer aided manufacturing (CAM), trajectory generation, feed-rate generation, axis servo control loop, and error estimation. However, deformations of the mechanical structure were not considered.

The virtual machine tool, as described by Altintas et al. [2], comprises mechanical structure, control loop, and manufacturing process. For the combined simulation of these aspects, they listed following methods: finite element (FE) simulation, rigid multi-body simulation (RMBS), and flexible multi-body simulation (FMBS). These modelling methods are reviewed in *Sections* 2.2, 2.3, and 2.4, respectively.

Furthermore, in order to be able to use FE models in FMBS, the models have to be reduced to low-order approximations. The state of the art in model reduction is reviewed in *Section* 2.5.

Modelling of the interfaces to flexible bodies is discussed in *Section* 2.6 and applications in the field of machine tool simulation are outlined in *Section* 2.7.

The subsequent discussion of the state of the art in *Section* 2.8, respecting the aim of the thesis, leads to a selection of methods and to a research gap, which is presented in *Section* 2.9 and shall be filled with the thesis at hand.

2.2 Finite Element Simulation

Fundamentals of the finite element method can be found, e.g. in the textbooks of Knothe and Wessels [56], Bathe [10], Rao [83], Steinke [104], and Schwertassek and Wallrapp [95]. Through discretisation of the domain using elements of finite dimension, simple shapes, and polynomial Galerkin interpolation functions, the governing partial differential equations (PDEs) can be approximated through systems of ordinary differential equations (ODEs), differential algebraic equations (DAEs), or algebraic equations.

In the case of structural dynamics problems with linear elastic material properties, finite element discretisation leads to the system of ODEs

$$\boldsymbol{M}\,\ddot{\boldsymbol{x}} + \boldsymbol{D}\,\dot{\boldsymbol{x}} + \boldsymbol{K}\,\boldsymbol{x} = \boldsymbol{f} \tag{2.1}$$

with the mass matrix $\boldsymbol{M} \in \mathbb{R}^{N\times N}$, the damping matrix $\boldsymbol{D} \in \mathbb{R}^{N\times N}$, the stiffness matrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}$, the displacement vector $\boldsymbol{x} \in \mathbb{R}^{N}$, and the total load vector $\boldsymbol{f} \in \mathbb{R}^{N}$. The finite element discretisation leads to system matrices with following important properties:

- the system matrices are symmetric [104],
- the system matrices are sparse [104],
- the stiffness matrix is positive semi-definite, whereas the rank drop corresponds to the number of rigid body degrees of freedom [56], and
- the mass matrix is positive definite [104].

As pointed out by Maglie [71], the system matrices of complex machine tools are usually of large dimension and can reach up to several million degrees of freedom, and thus the system of *Eq.* (2.1) has to be reduced to a system of lower order, which approximates the original system accurately enough.

2.3 Rigid Multi-Body Simulation

Powerful formalisms exist for the simulation of rigid multi-body systems. Shabana [98] as well as Schiehlen and Eberhard [92] used D'Alembert's principle to derive the Newton-Euler equations of motion for a rigid body according to

$$\boldsymbol{M}^{i}\,\ddot{\boldsymbol{q}}^{i} + \boldsymbol{C}^{i}_{\boldsymbol{q}}\,\boldsymbol{\lambda} = \boldsymbol{Q}^{i}_{\boldsymbol{e}} + \boldsymbol{Q}^{i}_{\boldsymbol{v}}, \qquad i = 1, 2, \ldots, n_b \tag{2.2}$$

In this equation, i denotes the number of the body in the multi-body system, n_b is the total number of bodies, $\boldsymbol{q}^i$ is the vector of generalised coordinates and $\boldsymbol{M}^i$ is the mass matrix of body i. Kinematic constraints between different bodies in the system are considered using the constraint Jacobian matrix $\boldsymbol{C}^{i}_{\boldsymbol{q}}$ and the vector of Lagrange multipliers $\boldsymbol{\lambda}$. The vector $\boldsymbol{Q}^{i}_{\boldsymbol{e}}$ is the vector of generalised external forces associated to the generalised coordinates of body i. $\boldsymbol{Q}^{i}_{\boldsymbol{v}}$ is called quadratic velocity vector and contains the gyroscopic and Coriolis force components.

According to Shabana [98], *Eq.* (2.2) can be separated into its translational and rotational components as

$$\begin{bmatrix} \boldsymbol{M}^{i}_{\boldsymbol{RR}} & \boldsymbol{M}^{i}_{\boldsymbol{R\theta}} \\ \text{symmetric} & \boldsymbol{M}^{i}_{\boldsymbol{\theta\theta}} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{R}}^{i} \\ \ddot{\boldsymbol{\theta}}^{i} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}^{i}_{\boldsymbol{R}} \\ \boldsymbol{C}^{i}_{\boldsymbol{\theta}} \end{bmatrix} \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{Q}^{i}_{\boldsymbol{e_R}} \\ \boldsymbol{Q}^{i}_{\boldsymbol{e_\theta}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{Q}^{i}_{\boldsymbol{v_R}} \\ \boldsymbol{Q}^{i}_{\boldsymbol{v_\theta}} \end{bmatrix}, \qquad i = 1, 2, \ldots, n_b\;. \tag{2.3}$$

Here, the index R denotes the rigid body translational, and θ the rotational degrees of freedom. The matrix $\boldsymbol{M}^{i}_{\boldsymbol{RR}}$ is a 3×3 constant diagonal matrix with the translational mass of body i on its diagonal. $\boldsymbol{M}^{i}_{\boldsymbol{\theta\theta}}$ and $\boldsymbol{M}^{i}_{\boldsymbol{R\theta}}$ denote the rotational components of the inertia matrix. In the case when the origin of the rigid body reference system is attached to the centre of mass, the matrix $\boldsymbol{M}^{i}_{\boldsymbol{\theta\theta}}$ is the inertia tensor of the body and $\boldsymbol{M}^{i}_{\boldsymbol{R\theta}}$ is zero.

2.4 Flexible Multi-Body Simulation

The rigid multi-body simulation formalisms presented in *Section* 2.3 underlie the assumption that the deformations of the bodies in a multi-body system are negligible. This is often not the case, as Shabana [98] stated, and therefore the compliance of the bodies has to be modelled.

In the case of small deformations and small rigid body movements, i.e. the translation or especially rotation of the body without deformation, linear finite element models, as described in *Section* 2.2, can be used to model the dynamic effects.

Shabana [97] showed that, in the case of large deformations and small rigid body movements, incremental finite element formulations can be used.

If the deformations of the bodies are small and the rigid body movements are large, the floating frame of reference formulation, as described, e.g. by Shabana [98], can be used. This method uses a finite set of shape functions, derived, e.g. with a Rayleigh-Ritz approximation or the finite element method, to describe the flexible behaviour of the bodies with a finite number of generalised coordinates $\boldsymbol{q}_f^i$. The position vector $\boldsymbol{r}$ to a point P on the body is then split into the global position vector to the origin of the reference frame $\boldsymbol{R}$ and the position vector $\boldsymbol{u}$ to the point P in local coordinates which is further split into the position vector in the undeformed state $\boldsymbol{u_0}$ and the displacement due to deformation $\boldsymbol{u_f}$. This representation is visualised in *Fig.* 2.1 and can be written as

$$\boldsymbol{r} = \boldsymbol{R} + \boldsymbol{A}\boldsymbol{u} = \boldsymbol{R} + \boldsymbol{A}\left(\boldsymbol{u_0} + \boldsymbol{u_f}\right) \tag{2.4}$$

with the rotation matrix $\boldsymbol{A}$ for rotation from the local to the global frame of reference. For the representation of a local vector in global coordinates, the notation

$$\bar{\boldsymbol{u}} = \boldsymbol{A}\,\boldsymbol{u} \tag{2.5}$$

is used.

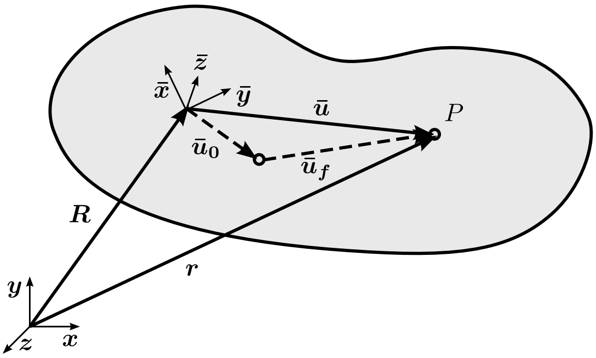


Figure 2.1: Representation of the floating frame of reference formulation. A bar on a letter denotes that the vector is transformed from local coordinates to global coordinates for display.

Another method that is shown by Shabana and Schwertassek [96] to be equivalent to the floating frame of reference formulation with finite element shape functions is the absolute nodal coordinates finite element formulation. In this method, the finite element coordinates are set up with respect to the inertial reference system.

Shabana [98] derived the equations of motion for the floating frame of reference formulation and the resulting system of equations is

$$\boldsymbol{M}^i\,\ddot{\boldsymbol{q}}^i + \boldsymbol{K}^i\,\boldsymbol{q}^i + \boldsymbol{C}_{\boldsymbol{q}}^i\,\lambda = \boldsymbol{Q}_e^i + \boldsymbol{Q}_v^i, \qquad i = 1, 2, \ldots, n_b\ , \tag{2.6}$$

which has a form similar to the equations of motion of a rigid multi-body system of *Eq.* (2.2). Here $\boldsymbol{K}^i$ denotes the body stiffness matrix. In terms of the translational coordinates $\boldsymbol{R}^i$, rotational coordinates

$\boldsymbol{\theta}^i$ and deformation coordinates $\boldsymbol{q}^i_f$, the equation can be written as

$$\begin{bmatrix} \boldsymbol{M}^i_{\boldsymbol{RR}} & \boldsymbol{M}^i_{\boldsymbol{R\theta}} & \boldsymbol{M}^i_{\boldsymbol{Rf}} \\ & \boldsymbol{M}^i_{\boldsymbol{\theta\theta}} & \boldsymbol{M}^i_{\boldsymbol{\theta f}} \\ \text{symmetric} & & \boldsymbol{M}^i_{\boldsymbol{ff}} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{R}}^i \\ \ddot{\boldsymbol{\theta}}^i \\ \ddot{\boldsymbol{q}}^i_f \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}^i_{\boldsymbol{ff}} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}^i \\ \boldsymbol{\theta}^i \\ \boldsymbol{q}^i_f \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}^i_{\boldsymbol{R}} \\ \boldsymbol{C}^i_{\boldsymbol{\theta}} \\ \boldsymbol{C}^i_{\boldsymbol{f}} \end{bmatrix} \boldsymbol{\lambda}$$

$$= \begin{bmatrix} \boldsymbol{Q}^i_{\boldsymbol{e_R}} \\ \boldsymbol{Q}^i_{\boldsymbol{e_\theta}} \\ \boldsymbol{Q}^i_{\boldsymbol{e_f}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{Q}^i_{\boldsymbol{v_R}} \\ \boldsymbol{Q}^i_{\boldsymbol{v_\theta}} \\ \boldsymbol{Q}^i_{\boldsymbol{v_f}} \end{bmatrix}, \qquad i = 1, 2, \dots, n_b \tag{2.7}$$

Here, the index f denotes the flexible degrees of freedom. Compared to *Eq.* (2.3), the coordinate vector has a new entry for the flexible degrees of freedom $\boldsymbol{q}^i_{\boldsymbol{f}}$ and the mass matrix has new submatrices representing the couplings between the elastic coordinates and the rigid body motion $\boldsymbol{M}^i_{\boldsymbol{Rf}}$ and $\boldsymbol{M}^i_{\boldsymbol{\theta f}}$ as well as the mass matrix of the elastic system $\boldsymbol{M}^i_{\boldsymbol{ff}}$. This shows that the inertia matrix depends on the elastic coordinates.

Moreover, the mass matrix $\boldsymbol{M}^i = \boldsymbol{M}^i(\boldsymbol{\theta}^i)$ depends on the orientation of the body, and thus the mass matrix has to be updated during the simulation process. In contrast, the stiffness matrix is constant for a linear elastic model of the deformation of the body.

Due to the large dimension of finite element models pointed out in *Section* 2.2, it is not practical to use the original system matrices in flexible multi-body simulation. Therefore, several authors proposed the use of model order reduction techniques in combination with the floating frame of reference formulation. Bauchau [12] and Schwertassek and Wallrapp [95] recommended modal reduction for the approximation of the system. Shabana [98] mentioned the component mode synthesis for the reduction of degrees of freedom. Lehner [63] and Nowakowski, Fehr, Fischer, and Eberhard [76], however, showed how to use further reduction methods such as Krylov subspace based reduction methods, balanced truncation, and proper orthogonal decomposition in combination with flexible multi-body simulation.

2.5 Model Order Reduction

2.5.1 Dynamic System

Considering the system from *Eq.* (2.1) and expressing the force input vector by the input matrix $\boldsymbol{B} \in \mathbb{R}^{N\times p}$ and the input signals $\boldsymbol{u} \in \mathbb{R}^p$, and calculating the output signals $\boldsymbol{y} \in \mathbb{R}^q$ using the output matrix $\boldsymbol{C} \in \mathbb{R}^{q\times N}$ and the state vector $\boldsymbol{x} \in \mathbb{R}^N$, one gets the system structure used, e.g. by Fehr [35], as follows

$$\begin{aligned} \boldsymbol{M}\,\ddot{\boldsymbol{x}} + \boldsymbol{D}\,\dot{\boldsymbol{x}} + \boldsymbol{K}\,\boldsymbol{x} &= \boldsymbol{B}\,\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\,\boldsymbol{x}\ . \end{aligned} \tag{2.8}$$

Many model reduction techniques and algorithms, however, are developed for linear systems of first-order structure, which can be expressed using the system matrices $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{E} \in \mathbb{R}^{N\times N}$ as

$$\begin{aligned} \boldsymbol{E}\,\dot{\boldsymbol{x}} &= \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\,\boldsymbol{x}\ , \end{aligned} \tag{2.9}$$

according to Schilders, Van der Vorst, and Rommes [93].

The system of *Eq.* (2.8) can, as Shabana [98] showed, be transformed to following first-order structure with order $2N$ using block matrices:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}} \\ \ddot{\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K} & -\boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \dot{\boldsymbol{x}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{u}$$

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \dot{\boldsymbol{x}} \end{bmatrix} , \tag{2.10}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{0}$ the zero matrix of appropriate dimension. Therefore, model reduction techniques for first-order systems can be used for second-order systems.

A special class of second-order systems are the proportionally damped systems as described, e.g. by De Silva [28]. These systems have a damping matrix proportional to the stiffness and mass matrix according to

$$\boldsymbol{D} = \alpha \boldsymbol{M} + \beta \boldsymbol{K} \ . \tag{2.11}$$

Proportional damping was introduced by Rayleigh [84] and is therefore also called Rayleigh damping. De Silva [28] showed that the proportionally damped and the undamped system possess identical real eigenvectors.

2.5.2 Projection-Based Model Order Reduction

Benner, Gugercin, and Willcox [17] listed three classes of model order reduction methods: data-fit reduction, projection-based reduction, and hierarchical reduction. All these methods derive reduced models from higher-order models. Data-fit models use input and output data of the original system to fit a lower order system, whereas projection-based methods use the original system operators for a projection in a subspace. Hierarchical reduced models are derived from higher-order models using approaches such as simplifying physics, coarser meshes, alternative basis expansions, and looser tolerances.

Benner et al. [17] also mentioned that the advantage of projection-based model reduction is that it preserves the structure of the underlying model. This is a mandatory property for the use with flexible multi-body simulation, and thus this thesis concentrates on projection-based model reduction techniques.

The projection framework for first-order systems according to *Eq.* (2.9) is described, e.g. by Benner, Mehrmann, and Sorensen [15]. The state coordinates $\boldsymbol{x}$ are projected onto a suitably defined low-dimensional trial subspace $\mathscr{V} = \mathrm{range}(\boldsymbol{V})$ by

$$\boldsymbol{x} = \boldsymbol{V} \hat{\boldsymbol{x}} \ , \quad \boldsymbol{V} \in \mathbb{R}^{N \times n} \ , \tag{2.12}$$

where $\mathrm{range}(\boldsymbol{V})$ denotes the subspace spanned by the columns of the matrix $\boldsymbol{V}$. This leads to the state coordinates of the reduced system $\hat{\boldsymbol{x}}$ and the system of equations

$$\boldsymbol{E} \boldsymbol{V} \dot{\hat{\boldsymbol{x}}} = \boldsymbol{A} \boldsymbol{V} \hat{\boldsymbol{x}} + \boldsymbol{B} \boldsymbol{u} + \boldsymbol{r} \tag{2.13}$$

with the residuum $\boldsymbol{r}$, which results from the approximation in a lower-dimensional subspace. As there are N equations but only n unknowns, *Eq.* (2.13) builds an overdetermined system of equations. By imposing orthogonality of the residuum to the test subspace $\mathscr{W} = \mathrm{range}(\boldsymbol{W})$, i.e.

$$\boldsymbol{W}^T \boldsymbol{r} = \boldsymbol{0} \ , \quad \boldsymbol{W} \in \mathbb{R}^{N \times n} \tag{2.14}$$

a unique solution can be found, as it is shown by, e.g. Eid [33]. This condition is called Galerkin orthogonality condition and leads to the reduced system of equations

$$\boldsymbol{W}^T \boldsymbol{E} \boldsymbol{V} \dot{\hat{\boldsymbol{x}}} = \boldsymbol{W}^T \boldsymbol{A} \boldsymbol{V} \hat{\boldsymbol{x}} + \boldsymbol{W}^T \boldsymbol{B} \boldsymbol{u}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{C} \boldsymbol{V} \hat{\boldsymbol{x}} \ . \tag{2.15}$$

These equations can be written as

$$\begin{aligned} \hat{\boldsymbol{E}}\,\dot{\hat{\boldsymbol{x}}} &= \hat{\boldsymbol{A}}\,\hat{\boldsymbol{x}} + \hat{\boldsymbol{B}}\,\boldsymbol{u} \\ \hat{\boldsymbol{y}} &= \hat{\boldsymbol{C}}\,\hat{\boldsymbol{x}} \end{aligned} \tag{2.16}$$

with the reduced system matrices

$$\hat{\boldsymbol{E}} = \boldsymbol{W}^T\boldsymbol{E}\boldsymbol{V}\,, \quad \hat{\boldsymbol{A}} = \boldsymbol{W}^T\boldsymbol{A}\boldsymbol{V}\,, \quad \hat{\boldsymbol{B}} = \boldsymbol{W}^T\boldsymbol{B}\,, \text{ and } \quad \hat{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{V} \tag{2.17}$$

and the approximate output coordinates $\hat{\boldsymbol{y}}$.

The projection

$$\boldsymbol{P} = \boldsymbol{V}\left(\boldsymbol{W}^T\boldsymbol{V}\right)^{-1}\boldsymbol{W}^T \tag{2.18}$$

is called Petrov-Galerkin projection or two-sided projection, and if $\boldsymbol{W} = \boldsymbol{V}$, the projection is called Galerkin-Projection or one-sided projection. From *Eq.* (2.12) and *Eq.* (2.14), it follows for the residuum

$$\boldsymbol{r} \perp \mathscr{V}\,, \qquad \boldsymbol{r} \perp \mathscr{W}\,, \tag{2.19}$$

i.e. the residuum is orthogonal to the subspaces $\mathscr{V}$ and $\mathscr{W}$, independent of the particularly chosen basis matrices $\boldsymbol{V}$ and $\boldsymbol{W}$, respectively. A basis matrix is a matrix which contains the basis vectors for a subspace as columns, i.e.

$$\boldsymbol{V} = \left[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\right]\,, \quad \boldsymbol{W} = \left[\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n\right] \tag{2.20}$$

for n basis vectors $\boldsymbol{v}_i$ and $\boldsymbol{w}_i$, respectively.

Similarly, according to Salimbahrami and Lohmann [89], the projected matrices of the second-order system *Eq.* (2.8) are

$$\begin{aligned} &\hat{\boldsymbol{M}} = \boldsymbol{W}^T\boldsymbol{M}\boldsymbol{V}\,, \quad \hat{\boldsymbol{D}} = \boldsymbol{W}^T\boldsymbol{D}\boldsymbol{V}\,, \quad \hat{\boldsymbol{K}} = \boldsymbol{W}^T\boldsymbol{K}\boldsymbol{V}\,, \\ &\hat{\boldsymbol{B}} = \boldsymbol{W}^T\boldsymbol{B}\,, \text{ and } \quad \hat{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{V}\,, \end{aligned} \tag{2.21}$$

which leads to the reduced second-order system

$$\begin{aligned} \hat{\boldsymbol{M}}\,\ddot{\hat{\boldsymbol{x}}} + \hat{\boldsymbol{D}}\,\dot{\hat{\boldsymbol{x}}} + \hat{\boldsymbol{K}}\,\hat{\boldsymbol{x}} &= \hat{\boldsymbol{B}}\,\boldsymbol{u} \\ \hat{\boldsymbol{y}} &= \hat{\boldsymbol{C}}\,\hat{\boldsymbol{x}}\,. \end{aligned} \tag{2.22}$$

Freund [43] showed that the transfer function of the reduced system is independent of the particular basis of a subspace, and thus reduction with any basis matrix $\tilde{\boldsymbol{V}} = \boldsymbol{V}\boldsymbol{L}$ and $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{L}$, with any non-singular matrix $\boldsymbol{L}$ of dimension $n \times n$, leads to the same transfer function.

Preservation of stability during model reduction is very important because it is possible to achieve a very accurately matching frequency response function even though the model is in fact unstable. In time domain, however, this unstable model may show unbounded results, as Bond [19] explained. Salimbahrami et al. [91] showed that there is, in general, no guarantee that stability of a system is preserved when using projection-based model reduction. However, as Freund [43] and Bond [19] showed, stability is preserved for first-order systems when using one-sided projection. Furthermore, Salimbahrami et al. [91] proved that symmetry and positive definiteness of the system matrices of second-order systems, and thus stability, are also preserved when using one-sided projection.

The construction of appropriate reduction bases is the task of the model reduction techniques. Different methods are described in the following sections.

2.5.3 Guyan Reduction

In 1956, Turner, Clough, Martin, and Topp [110] proposed a method for reduction of the stiffness matrix by elimination of coordinates at which no forces are applied. Guyan [48] extended this method in 1965 for the stiffness and mass matrix and created thereby the famous Guyan reduction method.

As Kim and Lee [54] showed, the Guyan method distinguishes between master (index 1) and slave degrees of freedom (index 2) and decomposes the stiffness problem

$$\boldsymbol{f} = \boldsymbol{K}\boldsymbol{x} \tag{2.23}$$

to

$$\begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} \\ \boldsymbol{K}_{12}^T & \boldsymbol{K}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} . \tag{2.24}$$

Assuming that the slave force vector $\boldsymbol{f}_2$ is zero, the slave displacement $\boldsymbol{x}_2$ can be written as

$$\boldsymbol{x}_2 = -\boldsymbol{K}_{22}^{-1}\boldsymbol{K}_{21}\boldsymbol{x}_1 \tag{2.25}$$

and the reduced stiffness matrix can be represented as

$$\hat{\boldsymbol{K}} = \boldsymbol{K}_{11} - \boldsymbol{K}_{12}\boldsymbol{K}_{22}^{-1}\boldsymbol{K}_{12}^T . \tag{2.26}$$

The original displacement vector $\boldsymbol{x}$ can then be approximated by

$$\boldsymbol{x} \approx \boldsymbol{V}\boldsymbol{x}_1 , \quad \boldsymbol{V} = \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{K}_{22}^{-1}\boldsymbol{K}_{12}^T \end{bmatrix} , \tag{2.27}$$

where $\boldsymbol{V}$ is the projection matrix for Guyan reduction. This projection matrix can be used to transform the stiffness matrix

$$\hat{\boldsymbol{K}} = \boldsymbol{V}^T\boldsymbol{K}\boldsymbol{V} \tag{2.28}$$

and the mass matrix

$$\hat{\boldsymbol{M}} = \boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V} . \tag{2.29}$$

As Guyan [48] already pointed out, the stiffness at the master degrees of freedom is modelled exactly, but the approximated mass matrix leads to approximated, and not exact, eigenvalues and eigenvectors. However, Kim and Lee [54] presented an accurate eigenvalue error estimator based on an iterative algorithm.

2.5.4 Component Mode Synthesis

2.5.4.1 Fundamentals

In 1965, Hurty [50] proposed the component mode synthesis (CMS) method, where the entire structure is divided into substructures (components). Each component is then represented using a set of suitably defined component modes, which are then used as a projection basis for a one-sided projection according to *Eq.* (2.21) of the second-order dynamic system from *Eq.* (2.8).

Hurty [50] proposed the use of fixed interface normal modes. Later, the use of further mode types was proposed. Subsequently, a brief description of the different mode types is given.

Rigid Body Modes According to Craig and Bampton [25], rigid body modes are the displacement vectors for rigid movement of a component, i.e. the movements in which the component is displaced without deformation.

Normal Modes Craig [27] classified the normal modes in following types: fixed interface normal modes, free interface normal modes, hybrid interface normal modes, and loaded interface normal modes.

Fixed interface normal modes can be obtained by solving the eigenvalue problem for the system with all interfaces fixed, free interface normal modes are the eigenvectors of the eigenvalue problem of the free system, hybrid interface normal modes are the eigenvectors of the system with a combination of fixed and free interfaces, and loaded interface normal modes are calculated from a system preloaded with inertia or stiffness at the interfaces.

Usually only the lower eigenfrequency modes, up to a frequency of interest, are retained, as Myklebust and Skallerud [74] explained. This was justified, e.g. by Dietz et al. [30], who states that the lower eigenmodes usually are more important than higher frequency modes. However, other strategies for mode selection were developed. Park and Park [79] presented a mode selection criterion, based on the residual flexibility attachment modes. A mode contribution criterion was defined by the summation of the contributions of a group of modes to the interface flexibility.

Constraint Modes Constraint modes were described by Craig [27] as the static deformation of a structure when a unit displacement is applied to one coordinate of a set of constraint coordinates while the remaining coordinates of that set are fixed and all other degrees of freedom of the structure are force-free.

Attachment Modes An attachment mode is the displacement vector of the structure for a unit force, in contrast to a unit displacement, applied to one coordinate of a set of constraint coordinates, while the remaining coordinates of that set are force-free, as described by Dietz and Knothe [29].

Residual Flexibility Attachment Modes Residual flexibility modes were introduced by Rubin [85] and are the residual contributions of forces to the displacement vector, if free normal modes are used.

Inertia Relief Modes Inertia relief modes were used by Craig and Hale [26] as addition to attachment modes for structures with rigid-body degrees of freedom. They are the static displacements of the component restrained at all boundary coordinates and loaded by inertia forces resulting from unit acceleration in each of the rigid-body modes.

Frequency Response Modes Dietz, Wallrapp, and Wiedemann [30] proposed the usage of frequency response modes, i.e. the forced oscillation shape for a particular harmonic load, to improve the frequency response accuracy of the reduced system.

Krylov Subspace Basis Craig and Hale [26] introduced block Krylov subspace bases instead of eigenvectors as component modes. Again, two different types were used: fixed interface and free interface block Krylov subspaces. See *Section* 2.5.5 for more about Krylov subspaces.

2.5.4.2 Variations of CMS

Some variations of the CMS method were presented in the past. Hurty [50] originally proposed the use of rigid body modes, constraint modes, and fixed interface normal modes. Craig and Bampton [25] modified Hurty's approach by using only constraint modes, and fixed interface normal modes. Rubin [85] proposed a method which results in the use of free interface normal modes, rigid body modes, and residual flexibility modes.

The use of free interface normal modes only is also called modal reduction and was sometimes used as benchmark for comparison of model reduction techniques, e.g. by Benner, Saak, Bonin, Zäh, Soppa, and Faßbender [16].

2.5.4.3 Error Estimation

Elssel and Voss [34], Kim and Lee [55], and Boo, Kim, and Lee [21] presented methods for a posteriori estimation of the relative eigenfrequency errors of the global (synthesised) structure for the Craig and Bampton method. Examples of Boo et al. [21] showed that the accuracy of the error estimators is very similar, and that the error can be underestimated by as much as a factor of 10.

Bathe and Dong [9] proposed a CMS enhancement with subspace iterations and presented a conservative eigenvalue error criterion for the global structure.

Jakobsson and Larson [51] presented an estimation for the error of the displacements in the reduced model measured in the energy norm.

2.5.5 Krylov Subspace Based Model Order Reduction

2.5.5.1 Fundamentals

Krylov subspace based model order reduction techniques are based on the projection into Krylov subspaces. Due to its property of matching moments of the transfer functions of the reduced and original system, Krylov subspace techniques are also called moment matching techniques, e.g. by Lohmann and Salimbahrami [68]. As Salimbahrami and Lohmann [89] pointed out, the moments are the coefficients of the Taylor series expansion of the transfer function.

Salimbahrami and Lohmann [88] defined a Krylov subspace of dimension n as

$$\mathscr{K}_n(\boldsymbol{A}, \boldsymbol{b}) = \operatorname{span}\left\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \ldots, \boldsymbol{A}^{n-1}\boldsymbol{b}\right\} \ . \tag{2.30}$$

It is spanned by the starting vector $\boldsymbol{b}$, iteratively multiplied by a matrix $\boldsymbol{A}$. These vectors build the basis of a Krylov subspace. Salimbahrami and Lohmann [88] proved that projection of the original system from *Eq.* (2.9) using basis matrices of the left and right Krylov subspaces $\boldsymbol{V}$ and $\boldsymbol{W}$, respectively, according to

$$\begin{aligned} \operatorname{range}(\boldsymbol{V}) &= \mathscr{K}_n\left(\boldsymbol{A}^{-1}\boldsymbol{E}, \boldsymbol{A}^{-1}\boldsymbol{b}\right) \\ \operatorname{range}(\boldsymbol{W}) &= \mathscr{K}_n\left(\boldsymbol{A}^{-T}\boldsymbol{E}^T, \boldsymbol{A}^{-T}\boldsymbol{c}^T\right) \ , \end{aligned} \tag{2.31}$$

leads to a reduced system with a transfer function which matches at least $2n$ moments of the original transfer function. Further, Salimbahrami and Lohmann [88] showed that, if a one-sided projection is used instead of a two-sided projection, e.g. by using the left or right Krylov subspace only, at least n moments are matched. Moreover, moments about an expansion point $s_0 \neq 0$ can be matched using the subspaces

$$\begin{aligned} \operatorname{range}(\boldsymbol{V}) &= \mathscr{K}_n\left((\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}, (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{b}\right) \\ \operatorname{range}(\boldsymbol{W}) &= \mathscr{K}_n\left((\boldsymbol{A} - s_0\boldsymbol{E})^{-T}\boldsymbol{E}^T, (\boldsymbol{A} - s_0\boldsymbol{E})^{-T}\boldsymbol{c}^T\right) \ . \end{aligned} \tag{2.32}$$

Bai [8] showed that two-sided Krylov subspace projection leads to Padé approximation of the transfer function. A Padé approximant is a rational function with numerator and denominator polynomials $\alpha_{n-1}(s)$ and $\beta_n(s)$, respectively, according to

$$\hat{G}(s) = \frac{\alpha_{n-1}(s)}{\beta_n(s)} = \frac{a_0 + a_1(s - s_0) + a_2(s - s_0)^2 + \ldots + a_{n-1}(s - s_0)^{n-1}}{1 + b_1(s - s_0) + b_2(s - s_0)^2 + \ldots + b_n(s - s_0)^n} \tag{2.33}$$

that matches the moments of $G(s)$ about s_0 as far as possible, i.e.

$$G(s) = \hat{G}(s) + \mathscr{O}\left((s - s_0)^{2n}\right) \ . \tag{2.34}$$

A combination of Krylov subspaces for multiple expansion points can be used for rational interpolation of the transfer function. This method is usually referred to as rational Krylov subspace method, e.g. by Grimme [46].

Salimbahrami and Lohmann [88] proved some important properties for Krylov subspace based reduction methods, which are restated here. First, the frequency response of the reduced order model is invariant to a change of the Krylov basis as long as it spans the same subspace. Further, they proved for two-sided Krylov subspace methods that the reduced system is invariant to changes of representation and realisation of the original system, i.e. premultiplying the system's state equation and the state vector, respectively, by an invertible matrix. This, however, is not the case for one-sided methods.

2.5.5.2 Multiple-Input Multiple-Output Systems

So far, Krylov subspace techniques for the reduction of single-input single-output (SISO) systems were presented. For multiple-input multiple-output (MIMO) systems, several Krylov subspace techniques were presented.

Grimme [46] proposed the combination of Krylov subspaces for each input and output vector for left and right Krylov subspaces, respectively, by adding all vectors to the projection bases. He showed that all transfer functions of the reduced system match the number of moments according to the SISO case.

Salimbahrami and Lohmann [88] showed, that for block Krylov subspaces, i.e. the subspace

$$\mathscr{K}_n(\boldsymbol{A},\boldsymbol{B}) = \operatorname{span}\left\{\boldsymbol{B}, \boldsymbol{A}\boldsymbol{B}, \boldsymbol{A}^2\boldsymbol{B}, \ldots, \boldsymbol{A}^{n-1}\boldsymbol{B}\right\} \ , \tag{2.35}$$

directly calculated with a starting matrix $\boldsymbol{B}$, in contrast to the starting vector $\boldsymbol{b}$ from *Eq.* (2.30), leads to the equal number of matched moments as the combination of Krylov subspaces. Freund [43] denoted the resulting transfer function as matrix-Padé approximant.

Gallivan, Vandendorpe, and Dooren [44] introduced a tangential interpolation approach. Instead of matching moments of the matrix-valued $(p \times q)$ transfer function of the system, this approach uses right and left tangential directions $\boldsymbol{r}_i$ and $\boldsymbol{l}_i$ to create the right and left Krylov subspaces, respectively. Thus, instead of using the input and output matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ for the construction of Krylov subspaces, a set of tangential directions $\boldsymbol{r}_i$ and $\boldsymbol{l}_i$ leads to a set of starting vectors

$$\begin{aligned} \boldsymbol{b}_i &= \boldsymbol{B}\boldsymbol{r}_i \\ \boldsymbol{c}_i &= \boldsymbol{C}^T\boldsymbol{l}_i \ . \end{aligned} \tag{2.36}$$

Therefore, additionally to a good selection of expansion points and the number of moments to match, suitable tangential directions are to be chosen. Druskin, Simoncini, and Zaslavsky [31] proposed an adaptive selection of expansion points and tangential directions for first-order tangential interpolation, which does not lead to optimal approximation but is reasonably efficient in terms of calculation time.

2.5.5.3 Optimal Approximation

Optimality in system approximation using Krylov subspaces was addressed in several works in the last decade. The idea is, according to, e.g. Flagg, Beattie, and Gugercin [42], to find a reduced system, which minimises the error of a frequency response function $H(i\omega)$ with respect to a specific signal norm, as e.g. the $\mathscr{H}_2$ norm according to

$$||H||_{\mathscr{H}_2} = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} |H(i\omega)|^2 d\omega} \tag{2.37}$$

Gugercin, Beattie, and Antoulas [47] presented the iterative rational Krylov algorithm (IRKA) for SISO systems which was shown by Flagg et al. [42] to be a locally convergent minimiser of the $\mathscr{H}_2$ optimization problem. It was shown that the first order optimality conditions, presented by Spanos, Milman, and Mingori [102], are fulfilled for first-order Krylov subspaces with expansion points at the mirror images of the eigenvalues of the system. In other words, the transfer function of the reduced system matches two moments at the mirror images of its poles. The eigenvalues are iteratively improved with the reduced system until convergence.

Van Dooren, Gallivan, and Absil [111] derived conditions for $\mathscr{H}_2$-optimality of MIMO systems using tangential interpolation and Kubalinska [58] presented a MIMO iterative rational interpolation algorithm (MIRIAm) which, based on Van Dooren et al. [111], iteratively uses the eigenvectors as tangential directions and chooses the expansion points such that the first-order optimality conditions are fulfilled. Xu and Zeng [117] studied the selection of interpolation points and tangential interpolation directions for $\mathscr{H}_2$-optimal approximation of the original transfer function based on the solution of Sylvester equations.

Wolf, Panzer, and Lohmann [116] showed that the first-order optimality condition can be relaxed to simple interpolation, i.e. matching one moment, of the transfer function at the mirror images of its poles for pseudo-optimal model reduction, and presented a new algorithm.

2.5.5.4 Systems with Second-Order Structure

For structure-preserving model order reduction of second-order systems according to *Eq.* (2.8), Salimbahrami and Lohmann [89] introduced second-order Krylov subspaces as

$$\mathscr{K}_{2,n}(\boldsymbol{A_1}, \boldsymbol{A_2}, \boldsymbol{b}) = \operatorname{span}\{\boldsymbol{p_0}, \boldsymbol{p_1}, \ldots, \boldsymbol{p_{n-1}}\} \ , \tag{2.38}$$

where the basis vectors $\boldsymbol{p_i}$ that span the Krylov subspace are defined by the one-step recurrence

$$\begin{aligned} \boldsymbol{p_0} &= \boldsymbol{b} \\ \boldsymbol{p_1} &= \boldsymbol{A_1}\boldsymbol{b} \\ \boldsymbol{p_i} &= \boldsymbol{A_1}\,\boldsymbol{p_{i-1}} + \boldsymbol{A_2}\,\boldsymbol{p_{i-2}} \ , \quad i = 2, 3, \ldots, n-1 \ . \end{aligned} \tag{2.39}$$

They proved that n moments are matched for Krylov subspaces of dimension n and one-sided projection, and $2n$ moments are matched for two-sided projection. This corresponds to the moment matching properties of first-order systems.

Eid, Salimbahrami, Lohmann, Rudnyi, and Korvink [32] showed that proportionally damped systems can be reduced using a one-sided first-order Krylov subspace developed without damping matrix, leading to the same properties as using second-order Krylov subspaces. Salimbahrami et al. [91] proved that the same holds for two-sided methods.

2.5.5.5 Error Estimation and Automation

An approximation error estimation is crucial for being able to automatically choose the reduction parameters for a model reduction algorithm. As Fehr [35] pointed out, the evaluation of the relative error between the transfer functions of the original and reduced systems are a good measure but, for large-scale systems, the computational costs for the evaluation of the transfer function of the original system is too high. Therefore, several methods for the estimation of the error were developed.

Bechtold, Rudnyi, and Korvink [14] presented two heuristic error estimation methods for Padé-type approximation of a SISO system. One heuristic is based on the observation that, for frequencies near the expansion point $s_0 = 0$, the relative error of the transfer function of a reduced system is nearly equal to the relative error calculated between two consecutive Padé orders. The second heuristic uses the observation that Hankel singular values of the reduced system converge to those of the original system. Using an extrapolation of the decay of the Hankel singular values, a boundary for the error can be estimated. These heuristics were presented with only one experimental result and the authors mention that more work is needed in terms of theoretical justification.

Grimme [46] presented an error estimation based on complementary reduced systems using two sets of expansion points. The relative error between these systems is calculated. Good results were reported for two sets with alternating expansion points. Grimme [46] developed a further error estimation approach based on residual expressions. It is pointed out that, although there are some good results in the experiments, these methods do not necessarily lead to good results near the poles of the transfer function.

Fehr [35] proposed an error estimator based on projection of the system into modal subspaces, partitioned by eigenfrequencies inside and outside an interesting frequency range. The error estimator was derived using the assumption that the eigenvalues and eigenvectors of the reduced system are exact inside the frequency range of interest, and upper bounds for some not computable terms. It was shown that the computational effort for the calculation of the error, although in a similar scale, is higher than for the reduction of the system. Furthermore, Fehr [35] presented a fully automated reduction algorithm with selection of expansion points at the frequency with maximum estimated error, automated selection of the number of moments to match, and automated stopping based on tolerances for the change of the estimated error.

Bonin et al. [20] presented a fully automated reduction algorithm for reduction of second-order systems with proportional damping. Starting from an initial set of expansion points, further expansion points, the reduced order, and the number of moments to match per expansion point are iteratively determined. The selection of expansion points is based on the eigenfrequencies of the reduced system, an adaptive selection of the expansion points to use is based on the error of the moments at the expansion points, and the stopping criterion is based on the $\mathscr{H}_2$ norm of the difference between the transfer functions of two consecutive systems.

2.5.5.6 Implementation of Krylov Subspace Basis Generation

Lanczos [60] introduced an iteration method for the solution of the eigenvalue problem. This process generates two sequences of basis vectors that span the left and right Krylov subspaces from *Eq.* (2.31). Aliaga, Boley, Freund, and Hernández [1] presented a Lanczos procedure for block Krylov subspaces. Unfortunately, as Salimbahrami, Lohmann, and Bechtold [90] stated, the Lanczos procedure is very limited in terms of numerical stability. In contrast, the Arnoldi procedure is said to be much more stable.

The Arnoldi algorithm, first proposed by Arnoldi [6] as an alternative to the Lanczos procedure, is commonly used for the creation of Krylov subspace bases. An algorithm suitable for model reduction is presented, e.g. by Salimbahrami et al. [90]. The output of the algorithm is an orthonormal basis for a Krylov subspace.

Salimbahrami and Lohmann [89] introduced a two-sided Arnoldi algorithm which, in fact, corresponds to executing the Arnoldi algorithm twice, once for each projection matrix $\boldsymbol{V}$ and $\boldsymbol{W}$, respectively.

An Arnoldi algorithm for the creation of second-order Krylov subspaces according to *Eq.* (2.38) was proposed by Bai, Meerbergen, and Su [7].

Salimbahrami and Lohmann [88] recommended to use the Arnoldi algorithm due to its simplicity in the implementation and numerical robustness.

The main calculation effort for both, the Lanczos as well as the Arnoldi procedure, for sparse system matrices lies in the repeated solution of large sparse systems of linear equations $(\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\,\boldsymbol{b}$ from *Eq.* (2.32), as Fischer and Eberhard [39] pointed out. Direct solvers provide the solution by LU decomposition. For each expansion point, the LU decomposition

$$\boldsymbol{L}\boldsymbol{U} = (\boldsymbol{A} - s_0\boldsymbol{E}) \tag{2.40}$$

has to be calculated and afterwards the systems of equations can be solved using forward and backward substitution.

Bai et al. [7] stated that the computational complexity for the construction of a Krylov subspace basis (*Eq.* (2.31)) using the Arnoldi procedure is $\mathscr{O}(N\,n^2)$ and thus, the calculation effort scales linearly with the order of the large-scale system. This makes the Arnoldi procedure suitable for systems of high order.

2.5.6 Balanced Truncation

One class of model reduction techniques is based on balancing a system in terms of input and output energy, and truncating the least important states. The short introduction given by Witteveen [115] is used here, based on a system according to *Eq.* (2.9) with $\boldsymbol{E} = \boldsymbol{I}$. The system with the controllability Gramian

$$\boldsymbol{W_c}(T) = \int_0^T e^{-\boldsymbol{A}t}\boldsymbol{B}\boldsymbol{B}^T e^{-\boldsymbol{A}^T t}\,dt \tag{2.41}$$

and the observability Gramian

$$\boldsymbol{W_o}(T) = \int_0^T e^{\boldsymbol{A}^T t}\boldsymbol{C}^T\boldsymbol{C}e^{\boldsymbol{A}t}\,dt \tag{2.42}$$

is called balanced when the Gramians are diagonal matrices and $\boldsymbol{W_c} = \boldsymbol{W_o}$. The diagonal elements of the Gramians are the so called Hankel singular values σ_i. A large Hankel singular value means that the output energy is strongly influenced by this state, and that the input energy needed to control the particular state is high. Model order reduction can be performed by truncating the states with the lowest Hankel singular values, and thus the lowest importance.

The controllability and observability Gramians can be calculated by solving following Lyapunov equations

$$\boldsymbol{A}\boldsymbol{W_c} + \boldsymbol{W_c}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T = 0 \tag{2.43}$$

$$\boldsymbol{A}^T\boldsymbol{W_o} + \boldsymbol{W_o}\boldsymbol{A} + \boldsymbol{C}^T\boldsymbol{C} = 0\,. \tag{2.44}$$

As e.g. Batlle and Roqueiro [11] pointed out, the Gramians are symmetric, positive definite matrices and thus have a Cholesky decomposition

$$\boldsymbol{W_c} = \boldsymbol{X}\boldsymbol{X}^T \; and \quad , \boldsymbol{W_o} = \boldsymbol{Y}\boldsymbol{Y}^T \tag{2.45}$$

and computing the SVD

$$\boldsymbol{Y}^T\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V} \tag{2.46}$$

gives the matrix $\boldsymbol{\Sigma}$ with the Hankel singular values on its diagonal. The balancing transformation basis is then given by

$$\boldsymbol{T} = \boldsymbol{X}\boldsymbol{V}\boldsymbol{\Sigma}^{-\frac{1}{2}}\,. \tag{2.47}$$

As Antoulas [5] states, the computational complexity for the solution of the Lyapunov equations and the singular value decomposition is $\mathcal{O}\left(N^3\right)$. Su, Balakrishnan, and Koh [106] presented a slightly more efficient method for balanced truncation using Krylov subspace methods for the computation of low-rank approximations of the Gramians. Nevertheless, they state the same asymptotic computational complexity as for the standard method. However, Chahlaoui [23] presented algorithms for the calculation of approximations for the Gramians with a computational cost $\mathcal{O}(N)$ by properly making use of the sparsity of the system matrices.

Witteveen [115] mentions that balanced truncation, due to the truncation of some states, leads to inexact static behaviour of the reduced systems. This can also be seen in the examples of Benner, Saak, Bonin, Zäh, Soppa, and Faßbender [16].

2.5.7 Proper Orthogonal Decomposition

Another common approach for model reduction is proper orthogonal decomposition (POD), also known as Karhunen-Loeve decomposition. Pinnau [80] described model reduction by means of POD as the task to find a subspace which approximates a set of data, also called snapshots, in an optimal least squares sense. As Schilders et al. [93] points out, the idea is that the time-response of a system on a certain input contains the essential behaviour of the system. Using singular value decomposition (SVD) of the snapshot matrix, containing the snapshots as columns, the basis for projection can be calculated.

2.6 Coupling of Flexible Bodies

2.6.1 Modelling of Fixed Interfaces

In structural mechanics finite element modelling, the degrees of freedom are the nodal coordinates, and thus, forces are applied to those nodal coordinates as, e.g. Knothe and Wessels [56] showed. Therefore, for realistic modelling of the interaction with a flexible body, one has to define a meaningful distribution of the interface force to the nodes.

In context of machine tools, the interfaces are needed in order to model guiding and driving components like linear guides, bearings, mounting elements for the machine basis, or ball screws, as listed by Maglie [71]. Usually, according to Siedl [99], the components do not need to be modelled with all details. E.g. for a linear guide, it is not necessary to model the contacts between the rolling elements (balls, cylinders) and the rolling surface. Baudisch [13], Zaeh, Oertli, and Milberg [118], Neithardt [75], Siedl [99], and Maglie [71] modelled the connection between both sides of the coupling components by connecting each surface of the coupling component to a reference node either by kinematic constraints or by force distribution. Therefore, force distributing elements were used, which are provided by the finite element software. Multi-directional spring and damper elements were used to connect the newly created reference nodes using stiffness matrices. Maglie [71] researched the difference between the rigid elements and force distribution in detail. A comparison of simulations conducted using catalogue values for the stiffness of linear guides with measurement results lead to the conclusion that force distribution elements should be used in order to avoid undesirably stiffening of the structure.

Despite coupling elements in mechanical engineering are always elastic, coupling using algebraic constraints using, e.g. Lagrange multipliers is possible. However, Simeon [101] investigated the use of Lagrange multipliers in flexible multi-body dynamics and concluded that Lagrange multipliers lead to ill-conditioned systems.

2.6.2 Modelling of Moving Interfaces

In order to model moving loads and couplings on flexible bodies, such as linear guides, ball screw spindles, or measurement systems, one has to be able to model interfaces with changing position of action on the body. Several approaches were developed to handle this problem.

2.6.2.1 Direct Nodal Methods

Fischer and Eberhard [37] showed an example of simulation of the turning process on a thin-walled cylinder. The interaction between the tool and the workpiece moves around the shoulder of the cylinder, which is discretised using the finite element method and reduced using modal reduction. The interaction is directly applied on the nodes and interpolated using bilinear interpolation between the nodes.

Siedl [99] proposed the use of node weighting with a piecewise parabolic weighting function spanning at least five nodes on an equidistantly meshed line. This method was shown to eliminate oscillations induced by the discretisation using a finite element mesh almost entirely.

2.6.2.2 Reduction of the Interface Matrix

Stykel and Vasilyev [105] presented a method for approximation of the interface matrices to matrices of lower rank in order to reduce the number of in- and outputs of the system. This is important in combination with model order reduction using techniques which are dependent on the interface matrices. The goal of the proposed method is to approximate the columns of the input matrix $\boldsymbol{B}$ from *Eq.* (2.8), consisting of one column for each node on the moving path, in a lower dimensional subspace. The idea is to approximate the input to the system as

$$\hat{\boldsymbol{u}}(t) \approx \boldsymbol{\Psi}(t)\boldsymbol{u}(t)\,, \tag{2.48}$$

using a time dependent vector valued function $\boldsymbol{\Psi}(t)$, which leads to an approximated input matrix $\hat{\boldsymbol{B}}$ satisfying

$$\boldsymbol{B}(t) \approx \hat{\boldsymbol{B}}\boldsymbol{\Psi}(t)\ . \tag{2.49}$$

Stykel and Vasilyev [105] derived the error bound for the system output due to the input approximation and used this to formulate a linear least squares problem that minimises the $\mathscr{L}_2$-norm error for a given function $\boldsymbol{\Psi}$. As choice for $\boldsymbol{\Psi}$, Stykel and Vasilyev [105] proposed the use of either Legendre polynomials, B-splines (see, e.g. Schumaker [94]), or coarse finite element interpolation functions (see, e.g. Rao [83]). The method was tested for a moving point load on a one-dimensional Euler-Bernoulli beam model and Fischer, Vasilyev, Stykel, and Eberhard [41] successfully tested the method for a moving point load on a rotating thin-walled cylinder.

2.6.2.3 Static Mode Switching

Based on the assumption that albeit many degrees of freedom of a system can be loaded during simulation, few are loaded simultaneously, Heirman, Tamarozzi, and Desmet [49] introduced a static mode switching method in combination with a component mode synthesis method. The idea is to use a constant set of free interface normal modes and, depending on the state of loading, a changing set of attachment modes (see *Section* 2.5.4). This leads to lower-order systems during simulation than using component mode synthesis with all possible attachment modes. A problem of this method is, as Heirman et al. [49] pointed out, that removing of a mode from the mode set and thus setting a state abruptly to zero introduces a discontinuity during transient simulation. To deal with this, numerical damping for higher frequencies was proposed.

Tamarozzi et al. [107] adopted the static mode switching method for contact simulation and showed its efficiency with an example of a gear contact simulation. However, the problem of discontinuities was still not solved.

2.6.2.4 Parametric Model Order Reduction

As Fischer and Eberhard [38] showed, systems with moving loads can be seen as linear time-variant systems with parameter dependence. For this purpose, the input matrix $\boldsymbol{B}$ which contains a large number of columns $\boldsymbol{b}_i$, namely one for each of the k possible nodal forces, is replaced by a parameter dependent matrix $\boldsymbol{B_e}(p)$ according to

$$\boldsymbol{B_e}(p) = \sum_{i=1}^{k} w(p)\,\boldsymbol{b}_i \tag{2.50}$$

with the parameter p and the weighting functions w_i. The output matrix $\boldsymbol{C}$ can be treated equivalently.

Fischer and Eberhard [38] proposed to derive k reduced order systems into a consistent set of generalised coordinates, one for each input vector, and then either interpolate the systems using matrix-interpolation as shown by Panzer, Mohring, Eid, and Lohmann [78] or by interpolating the manifolds as shown by Amsallem and Farhat [3]. This interpolation is performed on-line during simulation.

Examples presented by Fischer and Eberhard [40], Fischer et al. [41], or Lang, Saak, and Benner [61] show the applicability of the method for problems of moderate complexity.

2.7 Machine Tool Simulation

Several works were published, which address the simulation of machine tools using some of the methods mentioned above.

Weikert [114] presented an application oriented simulation of machine tools and its components, whereas the structure was modelled as an assembly of rigid bodies.

Berkemer [18] investigated the coupled simulation of structural mechanics, drives, and control loop of machine tools using flexible multi-body simulation. For the reduction, component mode synthesis methods were proposed.

Baudisch [13] presented a high level simulation method as a strategy for data and model management in simulation of machine tools during development. Again, flexible multi-body simulation was used in combination with component mode synthesis model reduction. Commercial tools like SIMPACK, Matlab and Simulink were used for the simulation.

Neithardt [75] researched methods for simulation and optimisation of machine tools during the concept and design phase. During the concept phase, optimisation of design parameters using rigid multi-body simulation was proposed, and during the design phase, flexible multi-body simulation in combination with component mode synthesis was proposed. The simulation was accomplished using the commercial software tools MSC.Nastran and MSC.Patran for finite element simulation, MSC.Adams for flexible multi-body simulation, and OptiSLang for parameter optimisation.

Siedl [99] investigated simulation methods for the dynamic behaviour of machine tools during axis movements. For the first time, the problem of moving interfaces and effects of oscillation due to finite element discretisation was addressed, as discussed in *Section* 2.6.2. For the simulation, the commercial flexible multi-body dynamics software RecurDyn was used in combination with MSC.Nastran and MSC.Patran. A combination of reduced models derived using component mode synthesis and large-scale models was used. The bodies which have moving interfaces were not reduced.

Oertli [77] examined the structural calculation and controller simulation of machine tools with focus on modelling of machine components like drive shafts, bearings, clutches, linear guides, ball screws, and gears. Modal reduction of MSC.Nastran/MSC.Patran models was used.

Maglie [71] introduced methods for the systematic simulation of machine tools during the design phase. Maglie [71] extended the axis construction kit (ACK), presented by Lorenzer, Weikert, and Wegener [69], for the use with reduced finite element models. For model reduction, the block Arnoldi implementation MOR for ANSYS, presented by Rudnyi and Korvink [86], was used. This tool did not feature any error estimation. The reduced models were used for simulation with Matlab and Simulink.

Wabner, Frieß, Hofmann, Hellmich, and Quellmalz [112] presented a concept for simulation parallel to the design process. A virtual prototype was constructed for the assistance during design and controller parametrisation. ANSYS finite element models were reduced using modal reduction and used for simulation with Matlab.

Moreover, a few specific applications of virtual prototypes for simplified simulation and estimation of machine tools performance were introduced. Weikert, Jaeger, Bossoni, and Wegener [113] proposed a method for estimation of controller parameters and calculation of closed-loop transfer functions using rigid multi-body simulation and the ACK. Transient simulations were used for the evaluation of deviations at the tool centre point.

Kono, Weikert, Matsubara, and Yamazaki [57] presented a simple evaluation of the mechanical dynamic error for machine tool structures, again using ACK, and compared this with the results from full finite element simulation. Quasi-static simulation with inertial loads for the acceleration load case was performed.

2.8 Discussion of the State of the Art

In this section, the presented state of the art is discussed and challenged with the requirements on a simulation method for machine tools presented in *Chapter* 1. Where possible, appropriate methods are chosen and, otherwise, the research gap is identified.

2.8.1 Evaluation of Simulation Methods

Linear finite element approaches do not support large rotations of the bodies and are thus not appropriate as standalone method. Rigid multi-body simulation, however, feature the ability to simulate large rotations but not the structural deformations. Flexible multi-body simulation allows both, the simulation of large rotations as well as structural deformations. Due to small, linear-elastic deformations, the floating frame of reference formulation with linear-elastic finite element modelling of the elasticity is adequate for the purpose of simulation of machine tools.

2.8.2 Evaluation of Model Order Reduction Methods

The finite element models used for the flexible multi-body simulation have a large number of degrees of freedom and can thus not be used for efficient simulation in practice. Therefore, an appropriate model order reduction has to be used. Versatile model order reduction techniques were presented.

In order to preserve the structure of the model, projection-based methods suite best. Although models with second-order structure could be converted to first-order structure in order to use first-order methods, the second-order structure has to be preserved in order to be compatible with the floating frame of reference formulation. However, for proportionally damped structures, Krylov and eigenvector bases can equivalently be generated using first-order methods and ignoring the damping matrix. Thus, proportional damping should be used as model for material damping of the single bodies. Because the mechanical models of the single bodies are reduced separately and the couplings are then established between the reduced models, the damping in the couplings can be chosen arbitrarily during assembly of the system.

Guyan's method is not practical because all nodal degrees of freedom at the interfaces have to be retained in the reduced model. This leads to large reduced models, if not only single nodes are used as interfaces. Moreover, an error estimation for the eigenfrequency errors only exist, but one is interested in the maximum error of the frequency response.

Component mode synthesis is a promising reduction method. Error estimators for the energy norm of the transfer function as well as for the eigenvalues exist but, again, no estimator for the maximum error of the transfer function was developed so far.

Krylov subspace projection methods are promising in terms of flexibility and accuracy. However, there is no method which satisfies all posed criteria. The methods which lead to $\mathscr{H}_2$-optimal approximation, e.g. IRKA or MIRIAm, are not efficient in terms of calculation, because they require a multitude of expansion points, and for each expansion point, an LU-decomposition of the large-scale system has to be calculated. Furthermore, for machine tool simulation, it is not important to achieve $\mathscr{H}_2$-optimality but to achieve low maximum errors in a predefined frequency range.

Tangential interpolation methods lead to reduced systems of lower order than block Krylov methods. However, they do not use the exact inputs and outputs of the system but a linear combination of those. Thus, the static behaviour is not matched exactly. Matching of the static stiffness is crucial for static and quasi-static analyses, and thus has to be achieved.

The presented error estimators for Krylov based reduction seem to work well for detection of convergence in iterative algorithms, but they are not very accurate for low order approximations with loose error limits. The error estimator presented by Bechtold et al. [14] lacks on provable error bounds and already the authors mentioned that it is uncertain if the method is applicable for general problems. The suggestions of Grimme [46] and Fehr [35] have high computational costs due to the multitude of required expansion points and the latter relies on a high level of assumptions. Moreover, the repeated evaluation of the transfer function matrices of the reduced systems for multiple frequencies, in order to calculate the error estimation, is a computational intensive task, especially for systems with many inputs and outputs.

The most recently presented reduction algorithms with application in the field of flexible multi-body simulation, proposed by Fehr [35] and Bonin et al. [20], both lead to minimal error and not to an error below a definable tolerance and are computationally inefficient due to the use of many expansion points.

In order to preserve stability of the original model, only one-sided methods should be used. On the contrary, two-sided Krylov subspace projection should be used in order to get systems which are invariant to realisation and representation of the large-scale model. Both of these conditions can be fulfilled only if the input and output matrices are equal. This can be achieved by using each input as an output and vice versa.

In terms of implementation of Krylov algorithms, the Arnoldi algorithm should be used due to better numerical stability and more flexibility in implementation.

Further, proper orthogonal decomposition needs transient simulation for generation of the snapshots of the original system, which is infeasible for the large-scale systems addressed with this thesis.

Balanced truncation methods lead to inexactly matched static behaviour due to the truncation of states without substitution. Furthermore, they are not applicable for large-scale systems due to their computational complexity of $\mathcal{O}\left(N^3\right)$.

To summarise, there is no method which is adequate for order reduction of models applicable for flexible multi-body simulation of machine tools. There is a need for a method which is efficient in terms of calculation and reduction. Furthermore, there is no method that allows efficient estimation of error bounds in order to be able to ensure accurate results within a certain frequency range.

2.8.3 Evaluation of Interface Modelling Methods

All the interface definitions shall rely on persistent topology data as, e.g. faces, edges, and vertices of the geometrical model and not on finite element mesh data like nodes or elements. When referencing mesh nodes directly, the assignment is lost after modification of the mesh, e.g. due to mesh refinement or geometry changes, and this has to be avoided in order to maintain an efficient work-flow in the modelling process.

For modelling of fixed interfaces, utilising single nodes of a finite element model is not appropriate for modelling of machine tools. The stiffness at a single node depends on the size of the associated element and thus potentially changes when modifying the mesh.

Force distributing elements provided by the finite element software may be technically appropriate for the modelling of fixed interfaces but suboptimal in terms of the desired straightforward work-flow. The interfaces should be defined after export from the finite element software in order to avoid iterations over the finite element software and export process. Therefore, a new implementation of force distributing interfaces for forces and torques shall be developed.

Some methods for handling of moving interfaces were presented. Direct nodal methods which require an interface to the system for each nodal degree of freedom are not appropriate in combination with model order reduction, because the reduced order is highly dependent on the number of interfaces.

The reduction of the interface matrix according to Stykel and Vasilyev [105] needs a priori knowledge about the transient input and is thus not practical.

Mode switching methods for modelling of moving interfaces with component mode synthesis introduces discontinuities and leads to unsolved numerical problems during calculation.

Parametric model reduction requires multiple reductions of the large-scale system for different parameter values and, therefore, is computationally costly. Moreover, the on-line interpolation between the system matrices or subspaces could lead to inefficient simulation.

The latter three methods are designed for application of moving loads to only one node at a time. This is not necessary for the simulation of machine tool components which always have a force application in an area with some extent.

Finally, no methods for the application of torques on moving interfaces have been found in literature.

2.8.4 Review of Applications of Machine Tools Simulation

In the context of machine tools, no application for virtual prototypes was found, which uses reduced order models with a controlled error. All applications which use finite element models define the interfaces on a nodal basis which, as mentioned before, leads to unwanted iteration loops in the modelling work-flow.

In terms of evaluation of the machine performance, there is a need for a simple and systematic method which allows the efficient calculation of the dynamic accuracy of feed axes.

2.9 Research Gap and Action Plan

From the study of the state of the art and the goal of the thesis, one can draw following conclusions. There is a research gap incorporating following points:

- An efficient model order reduction with a predefined error bound within a specified frequency range.
- A method for modelling of moving interfaces which can be applied together with the model order reduction method from the previous point.
- A method for modelling of moving interfaces which allows the application of torques.
- Methods for efficient evaluation of the dynamic performance of a machine tool using a virtual prototype and frequency-domain analysis without the need for time-consuming transient simulations.
- A complete framework for the efficient and accurate simulation of the dynamics of machine tools comprising the aforementioned points.

This leads to following tasks for this thesis.

Development of a Model Order Reduction Method Force spectra of loads applied to machine tools shall be characterised and a frequency range of interest shall be determined. A model order reduction method has to be developed, which enables the reduction of large-scale models. The method shall feature a predefined error bound for the transfer functions inside a specified frequency range, but the static behaviour shall be matched accurately.

Either a Krylov subspace based method or a component mode synthesis method are to be used as a one-sided projection method with equal sets of inputs and outputs in order to preserve stability and structure of the system.

Development of Methods for Interface Definition Methods for the mesh-independent definition of interfaces to persistent topologies of the geometric model, e.g. faces, shall be developed, which can be applied after the export from the finite element software.

A force-distribution implementation for fixed interfaces and a new method and implementation for modelling of moving interfaces on flexible bodies which is compatible with model order reduction shall be developed.

All interfaces should enable to apply forces/translations and torques/rotations.

Implementation of a Flexible Multi-Body Simulation Environment A flexible multi-body simulation environment which is tailored for the simulation of machine tools and is capable to deal with the afore mentioned reduced order models and the moving interface description has to be implemented. The floating frame of reference formulation shall be used.

Development of a Framework for an Efficient Modelling Work-Flow All methods mentioned above are to be integrated in a framework and a software solution for the simulation of machine tools shall be developed in order to create an efficient modelling work-flow.

Development of an Efficient Method for the Evaluation of Dynamic Performance of Machine Tools New methods for the efficient evaluation of the dynamic performance of a machine tool are to be developed. The methods shall allow the calculation of maximum contour errors for feed axes following dynamic trajectories. Frequency-domain calculations of the system linearised at arbitrary positions in the workspace should be used in contrast to time-consuming transient simulations in order to enable efficient calculation of the dynamic performance of a machine tool.

Verification of the Methods in a Case Study The simulation and evaluation framework shall be tested by means of a case study.

2.10 Outline of the Thesis

In *Chapter* 3, a novel model reduction technique is presented which is based on projection into a combination of Krylov and modal subspaces. This is an efficient method in terms of the order of reduced systems as well as the calculation time. With an a-priori error bound estimation, the selection of reduction parameters is straightforward and an adjustable accuracy for a predefined frequency range can be achieved, while the static behaviour is matched accurately.

Chapter 4 discusses the formulation of interfaces, i.e. force distribution and displacement evaluation on flexible bodies. After introducing stationary interfaces, a novel method for modelling moving interfaces on flexible bodies is presented which is compatible with the presented model reduction method and allows the application of forces and torques respectively the evaluation of displacements and rotations.

In *Chapter* 5, based on a linearised floating frame of reference approach, an application oriented calculation of assembled system matrices for flexible bodies with an arbitrary orientation is presented. These matrices are then used together with the presented interface formulations for the creation of composed system matrices for an assembly of coupled flexible bodies. Moreover, formulations for simplified linear couplings for some important types of machine components like bearings, linear guides, and different types of gears are derived.

A software implementation of the presented methods for model order reduction, interface modelling, and assembly of flexible bodies is introduced in *Chapter* 6. The software package MORe fills the gap between a the finite element software ANSYS Mechanical [4] and the system simulation environment Simulink [100] as figured in *Fig.* 2.2.

Figure 2.2: Toolchain for modelling and simulation with MORe

In *Chapter* 7, a newly developed design process called Design to Specifications which enables an efficient application of simulation during the development of machine tool structures is presented. Besides techniques for efficient evaluation of the dynamic performance of machine tools, the design process introduces the derivation of requirements on controller bandwidth and structure eigenfrequencies from the specifications on accuracy and dynamic performance for a machine tool.

In *Chapter* 8, the presented methods are evaluated in a case study based on a gantry stage. A model of the gantry stage is validated by means of measurement results in time and frequency domain and the assumptions underlying the Design to Specifications process are proved useful.

Lastly, the results of the thesis are discussed in *Chapter* 9.

3 Model Order Reduction

After introducing the prerequisites in *Section* 3.1, the requirements on model order reduction for machine tools are discussed in *Section* 3.2. A novel model reduction method, utilising a combination of Krylov and modal subspaces, is presented in *Section* 3.3, and analysed in *Section* 3.4. In *Section* 3.5, a numerical implementation is presented and the usability is demonstrated by means of a real-world example in *Section* 3.6. The chapter is concluded with some final remarks on the model reduction method in *Section* 3.7.

3.1 Prerequisites

3.1.1 System Equations

The dynamic system with second-order structure was introduced in *Section* 2.5.1, according to

$$\begin{aligned} \boldsymbol{M}\,\ddot{\boldsymbol{x}} + \boldsymbol{D}\,\dot{\boldsymbol{x}} + \boldsymbol{K}\,\boldsymbol{x} &= \boldsymbol{B}\,\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\,\boldsymbol{x}\ . \end{aligned} \tag{3.1}$$

$\boldsymbol{M}, \boldsymbol{D}, \boldsymbol{K} \in \mathbb{R}^{N\times N}$ are the mass, damping, and stiffness matrices, respectively. $\boldsymbol{B} \in \mathbb{R}^{N\times p}$ is the input matrix corresponding to the input vector $\boldsymbol{u} \in \mathbb{R}^{p}$, and $\boldsymbol{C} \in \mathbb{R}^{q\times N}$ is the output matrix corresponding to the output vector $\boldsymbol{y} \in \mathbb{R}^{q}$. $\boldsymbol{x} \in \mathbb{R}^{N}$ is the displacement state vector.

In Laplace domain, the system equations can be rewritten as

$$\begin{aligned} s^2\,\boldsymbol{M}\,\boldsymbol{x} + s\,\boldsymbol{D}\,\boldsymbol{x} + \boldsymbol{K}\,\boldsymbol{x} &= \boldsymbol{B}\,\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\,\boldsymbol{x}\ . \end{aligned} \tag{3.2}$$

The transfer function of this system is

$$\boldsymbol{G}(s) = \boldsymbol{C}\,\left(\boldsymbol{M}\,s^2 + \boldsymbol{D}\,s + \boldsymbol{K}\right)^{-1}\,\boldsymbol{B} \tag{3.3}$$

and the corresponding frequency response function (FRF), found by setting $s = i\omega$, is

$$\boldsymbol{H}(i\omega) = \boldsymbol{G}(s)\big|_{s=i\omega} = \boldsymbol{C}\,\left(-\boldsymbol{M}\,\omega^2 + \boldsymbol{D}\,i\omega + \boldsymbol{K}\right)^{-1}\,\boldsymbol{B}\ . \tag{3.4}$$

Many operations can be carried out with the corresponding undamped system, described by

$$\begin{aligned} s^2\,\boldsymbol{M}\,\boldsymbol{x} + \boldsymbol{K}\,\boldsymbol{x} &= \boldsymbol{B}\,\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\,\boldsymbol{x}\ , \end{aligned} \tag{3.5}$$

by setting $\boldsymbol{D} = \boldsymbol{0}$. This system can be represented with first-order structure according to

$$\begin{aligned} \sigma\,\boldsymbol{E}\,\boldsymbol{x} &= \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\,\boldsymbol{x}\ . \end{aligned} \tag{3.6}$$

With $\sigma = s^2$, $\boldsymbol{E} = \boldsymbol{M}$, and $\boldsymbol{A} = -\boldsymbol{K}$, the transfer function of this system results in

$$\boldsymbol{G}(\sigma) = \boldsymbol{C}\,\left(\boldsymbol{E}\,\sigma - \boldsymbol{A}\right)^{-1}\,\boldsymbol{B}\ . \tag{3.7}$$

3.1.2 Modal Coordinates

The dynamic system can be transformed by congruent transformations into different coordinates without changing the system's input-output behaviour. A congruent transformation is a transformation of the form

$$\boldsymbol{B} = \boldsymbol{P}^T \boldsymbol{A} \boldsymbol{P} \tag{3.8}$$

with a non-singular matrix $\boldsymbol{P}$. A popular congruent transformation is the modal transformation, where the system is projected into modal coordinates. Therefore, the quadratic eigenvalue problem

$$\left(\boldsymbol{M}\, s_k^2 + \boldsymbol{D}\, s_k + \boldsymbol{K}\right) \boldsymbol{\phi}_k = \boldsymbol{0}\,, \quad k = 1, 2, \ldots, N \tag{3.9}$$

has to be solved for the eigenvalues s_k and eigenvectors $\boldsymbol{\phi}_k$. For the special case of $\boldsymbol{D} = \boldsymbol{0}$ and by substituting

$$s_k^2 = -\lambda_k\,, \tag{3.10}$$

the quadratic eigenvalue problem can be reformulated as generalised eigenvalue problem according to

$$(\boldsymbol{M}\, \lambda_k - \boldsymbol{K})\, \boldsymbol{\phi}_k = \boldsymbol{0}\,, \quad k = 1, 2, \ldots, N\,. \tag{3.11}$$

The matrix with all eigenvectors

$$\boldsymbol{\Phi} = \left[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N\right] \tag{3.12}$$

is called modal matrix. The eigenvectors are orthogonal to both, the mass and stiffness matrix, i.e.

$$\boldsymbol{\phi}_k^T \boldsymbol{M} \boldsymbol{\phi}_l = 0\,, \quad \boldsymbol{\phi}_k^T \boldsymbol{K} \boldsymbol{\phi}_l = 0\,, \quad \text{if } k \neq l\,. \tag{3.13}$$

From the positive definiteness of the mass matrix and the positive semi-definiteness of the stiffness matrix, it follows that

$$\lambda_k \geq 0\,, \quad \lambda_k \in \mathbb{R}\,, \quad k = 1, 2, \ldots, N\,. \tag{3.14}$$

Back-substituting the eigenvalues according to *Eq.* (3.10) leads to

$$s_k = \pm\sqrt{-\lambda_k} = \pm i\omega_k\,, \tag{3.15}$$

what shows that the eigenvalues s_k arise as purely imaginary, complex conjugate pairs of the eigenfrequencies ω_k.

The system can be projected with

$$\begin{aligned} \tilde{\boldsymbol{M}} &= \boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{\Phi}\,, & \tilde{\boldsymbol{D}} &= \boldsymbol{\Phi}^T \boldsymbol{D} \boldsymbol{\Phi}\,, & \tilde{\boldsymbol{K}} &= \boldsymbol{\Phi}^T \boldsymbol{K} \boldsymbol{\Phi}\,, \\ \tilde{\boldsymbol{B}} &= \boldsymbol{\Phi}^T \boldsymbol{B}\,, & \tilde{\boldsymbol{C}} &= \boldsymbol{C} \boldsymbol{\Phi}\,, \text{ and} & \tilde{\boldsymbol{x}} &= \boldsymbol{\Phi}^T \boldsymbol{x} \end{aligned} \tag{3.16}$$

into the modal space. Because of the $\boldsymbol{M}$- and $\boldsymbol{K}$-orthogonality of the modal matrix, the modal transformation diagonalises the mass and stiffness matrix, and thus, decouples the degrees of freedom of an undamped system. This is not the case for systems with arbitrary damping, but, for the special case of a proportionally damped system with $\boldsymbol{D} = \alpha\, \boldsymbol{M} + \beta\, \boldsymbol{K}$, the damping matrix will be transformed as

$$\begin{aligned} \tilde{\boldsymbol{D}} &= \boldsymbol{\Phi}^T \boldsymbol{D} \boldsymbol{\Phi} = \boldsymbol{\Phi}^T \left(\alpha \boldsymbol{M} + \beta \boldsymbol{K}\right) \boldsymbol{\Phi} = \alpha\, \boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{\Phi} + \beta\, \boldsymbol{\Phi}^T \boldsymbol{K} \boldsymbol{\Phi} \\ &= \alpha \tilde{\boldsymbol{M}} + \beta \tilde{\boldsymbol{K}}\,, \end{aligned} \tag{3.17}$$

which is again a diagonal matrix. For more information on diagonalisable quadratic eigenvalue problems, it is referred to Lancaster and Zaballa [59].

Usually, the eigenvectors are normalised to the mass matrix so that

$$\tilde{\boldsymbol{M}} = \boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{\Phi} = \boldsymbol{I} \ . \tag{3.18}$$

From *Eq.* (3.11), it follows that

$$\tilde{\boldsymbol{K}} = \operatorname{diag}(\lambda_k) = \operatorname{diag}\left(\omega_k^2\right) \ , \quad k = 1, 2, \ldots, N \ , \tag{3.19}$$

and from *Eq.* (3.17), that

$$\tilde{\boldsymbol{D}} = \operatorname{diag}(\alpha + \beta\lambda_k) = \operatorname{diag}\left(\alpha + \beta\omega_k^2\right) \ , \quad k = 1, 2, \ldots, N \ . \tag{3.20}$$

The transfer function from *Eq.* (3.4) can then be written as

$$\boldsymbol{G}(s) = \tilde{\boldsymbol{C}} \ \left(\boldsymbol{I}\, s^2 + \operatorname{diag}\left(\alpha + \beta\omega_k^2\right) s + \operatorname{diag}\left(\omega_k^2\right)\right)^{-1} \tilde{\boldsymbol{B}} \ , \quad k = 1, 2, \ldots, N \ , \tag{3.21}$$

or as a summation over the decoupled modes according to

$$\boldsymbol{G}(s) = \sum_{k=1}^{N} \frac{\tilde{\boldsymbol{C}}_{:k}\, \tilde{\boldsymbol{B}}_{k:}}{s^2 + \left(\alpha + \beta\omega_k^2\right) s + \omega_k^2} = \sum_{k=1}^{N} \frac{\tilde{\boldsymbol{C}}_{:k}\, \tilde{\boldsymbol{B}}_{k:}}{s^2 + 2\, \zeta_k\, \omega_k\, s + \omega_k^2} \ , \tag{3.22}$$

where $\boldsymbol{C}_{:k}$ is the k^{th} column vector of the matrix $\boldsymbol{C}$, $\boldsymbol{B}_{k:}$ is the k^{th} row vector of the matrix $\boldsymbol{B}$, and $\tilde{\boldsymbol{C}}_{:k}\, \tilde{\boldsymbol{B}}_{k:}$ is a dyadic product. The damping coefficient ζ_k for the k^{th} mode is, by comparison of coefficients,

$$\zeta_k = \frac{1}{2} \left(\frac{\alpha}{\omega_k} + \beta\, \omega_k \right) \ . \tag{3.23}$$

For undamped systems, the transfer function simplifies to

$$\boldsymbol{G}(s) = \sum_{k=1}^{N} \frac{\tilde{\boldsymbol{C}}_{:k}\, \tilde{\boldsymbol{B}}_{k:}}{s^2 + \omega_k^2} \ . \tag{3.24}$$

3.1.3 Properties of the System

3.1.3.1 Static Compliance

The dynamic compliance of the system corresponds to the FRF of *Eq.* (3.4). The static compliance $\boldsymbol{S}$ is the dynamic compliance, evaluated at $\omega = 0$ according to

$$\boldsymbol{S} = \boldsymbol{H}(i\omega)\big|_{\omega=0} = \boldsymbol{C}\, \boldsymbol{K}^{-1}\, \boldsymbol{B} \ , \tag{3.25}$$

or in modal coordinates

$$\boldsymbol{S} = \sum_{k=1}^{N} \frac{\tilde{\boldsymbol{C}}_{:k}\, \tilde{\boldsymbol{B}}_{k:}}{\omega_k^2} \ . \tag{3.26}$$

The representation in modal coordinates shows that all modes are involved in the static stiffness, provided that they are controllable and observable, i.e. $\tilde{\boldsymbol{C}}_{:k}\, \tilde{\boldsymbol{B}}_{k:} \neq \boldsymbol{0}$.

3.1.3.2 Poles of the Transfer Function

In the representation of the transfer function in modal coordinates in *Eq.* (3.22), the poles s_k are the zeros of the denominator

$$s^2 + 2\,\zeta_k\,\omega_k\,s + \omega_k^2\,, \tag{3.27}$$

which are

$$s_k = \left(-\zeta_k \pm \sqrt{\zeta_k^2 - 1}\right)\omega_k\,, \quad k = 1, 2, \ldots, N\,. \tag{3.28}$$

The poles of the undamped system are found by setting $\zeta_k = 0$ as

$$s_k = \pm i\omega_k\,, \quad k = 1, 2, \ldots, N\,. \tag{3.29}$$

These are complex conjugate pairs on the imaginary axis and correspond to the eigenvalues from *Eq.* (3.15).

3.1.3.3 Zeros of the Transfer Function

A zero in the transfer function, i.e. a transmission zero, is a specific Laplace parameter value s_{z_k} for which the output of a system according to *Eq.* (3.2) is zero. For the SISO system, i.e. $\boldsymbol{B} = \boldsymbol{b}$ and $\boldsymbol{C} = \boldsymbol{c}$, this can be written as

$$\begin{aligned} \left(\boldsymbol{M}\,s_{z_k}^2 + \boldsymbol{D}\,s_{z_k} + \boldsymbol{K}\right)\,\boldsymbol{\phi}_{z_k} - \boldsymbol{b}\,u_{z_k} &= \boldsymbol{0} \\ \boldsymbol{c}\,\boldsymbol{\phi}_{z_k} &= 0\,, \end{aligned} \tag{3.30}$$

where $\boldsymbol{\phi}_{z_k}$ is the oscillation shape and u_{z_k} is the corresponding input magnitude. This equation can be written using block matrices as

$$\left(\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{bmatrix} s_{z_k}^2 + \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{bmatrix} s_{z_k} + \begin{bmatrix} \boldsymbol{K} & \boldsymbol{b} \\ \boldsymbol{c} & 0 \end{bmatrix}\right) \begin{bmatrix} \boldsymbol{\phi}_{z_k} \\ u_{z_k} \end{bmatrix} = \boldsymbol{0} \tag{3.31}$$

which corresponds to a quadratic eigenvalue problem.

For undamped systems, *Eq.* (3.31) simplifies to a generalised eigenvalue problem according to

$$\left(\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{bmatrix} \lambda_{z_k} - \begin{bmatrix} \boldsymbol{K} & \boldsymbol{b} \\ \boldsymbol{c} & 0 \end{bmatrix}\right) \begin{bmatrix} \boldsymbol{\phi}_{z_k} \\ u_{z_k} \end{bmatrix} = \boldsymbol{0}\,, \tag{3.32}$$

with the eigenvalues $\lambda_{z_k} = -s_{z_k}^2$. All real positive eigenvalues λ_{z_k} lead to pure imaginary, complex conjugate Laplace parameters $s_{z_k} = \pm i\omega_{z_k}$. The first N elements of the corresponding eigenvectors, i.e. $\boldsymbol{\phi}_{z_k}$, are the oscillation shapes at the zero transmission frequencies ω_{z_k} zeros.

As an example, consider an undamped SISO-system in modal coordinates with the system matrices

$$\begin{aligned} &\tilde{\boldsymbol{M}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \tilde{\boldsymbol{K}} = \begin{bmatrix} \omega_1^2 & 0 \\ 0 & \omega_2^2 \end{bmatrix} = \begin{bmatrix} 100 & 0 \\ 0 & 400 \end{bmatrix}, \quad \tilde{\boldsymbol{b}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \tilde{\boldsymbol{c}} = \begin{bmatrix} 1 & 2 \end{bmatrix} \\ &\omega_1 = 10\,, \quad \omega_2 = 20\,. \end{aligned} \tag{3.33}$$

The transfer function of this system is according to *Eq.* (3.24)

$$H(s) = \sum_{k=1}^{2} \frac{\tilde{c}_k\,\tilde{\boldsymbol{b}}_k}{s^2 + \omega_k^2} = \frac{\tilde{c}_1\,\tilde{\boldsymbol{b}}_1}{s^2 + \omega_1^2} + \frac{\tilde{c}_2\,\tilde{\boldsymbol{b}}_2}{s^2 + \omega_2^2} = G_1(s) + G_2(s)\,. \tag{3.34}$$

The zeros can either be found by solving $G(s_z) = 0$ for the Laplace parameter s_z, or by solving the generalised eigenvalue problem from *Eq.* (3.32). The direct solution is applicable for systems of low order only, whereas the eigenvalue method is applicable for large systems too. The eigenvalues can be calculated by

$$\det\left(\begin{bmatrix}\tilde{\boldsymbol{M}} & \boldsymbol{0}\\ \boldsymbol{0} & 0\end{bmatrix}\lambda_z - \begin{bmatrix}\tilde{\boldsymbol{K}} & \tilde{\boldsymbol{b}}\\ \tilde{\boldsymbol{c}} & 0\end{bmatrix}\right) = \det\left(\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\lambda_z - \begin{bmatrix}\omega_1^2 & 0 & \tilde{\boldsymbol{b}}_1\\ 0 & \omega_2^2 & \tilde{\boldsymbol{b}}_2\\ \tilde{\boldsymbol{c}}_1 & \tilde{\boldsymbol{c}}_2 & 0\end{bmatrix}\right) = 0 \tag{3.35}$$

which leads to

$$\tilde{\boldsymbol{b}}_2\,\tilde{\boldsymbol{c}}_2\,\omega_1^2 + \tilde{\boldsymbol{b}}_1\,\tilde{\boldsymbol{c}}_1\,\omega_2^2 - \left(\tilde{\boldsymbol{b}}_1\,\tilde{\boldsymbol{c}}_1 + \tilde{\boldsymbol{b}}_2\,\tilde{\boldsymbol{c}}_2\right)\,\lambda_z = 0\,. \tag{3.36}$$

This equation can be solved for the eigenvalue according to

$$\lambda_z = \frac{\tilde{\boldsymbol{b}}_1\,\tilde{\boldsymbol{c}}_1\,\omega_2^2 + \tilde{\boldsymbol{b}}_2\,\tilde{\boldsymbol{c}}_2\,\omega_1^2}{\tilde{\boldsymbol{b}}_1\,\tilde{\boldsymbol{c}}_1 + \tilde{\boldsymbol{b}}_2\,\tilde{\boldsymbol{c}}_2} = \frac{1\cdot 400 + 2\cdot 100}{1+2} = 200 \tag{3.37}$$

and, since $\lambda_z = -s_z^2$, this leads to the zeros

$$s_{z1,2} = \pm\sqrt{s_z^2} = \pm\sqrt{-\lambda_z} = \pm\sqrt{-200} = \pm 10\sqrt{2}\,i\,. \tag{3.38}$$

The zeros are a complex conjugate pair on the imaginary axis, and thus, can be written as zeros in the FRF at

$$\omega_{z1,2} = \pm 10\sqrt{2} \tag{3.39}$$

by setting $s_{z_k} = i\omega_{z_k}$.

The zeros of the transfer function arise when the fractions G_1 and G_2 are expanded in order to attain a common denominator, and therefore, the zeros depend on the eigenvalues as well as on the in- and outputs of the system.

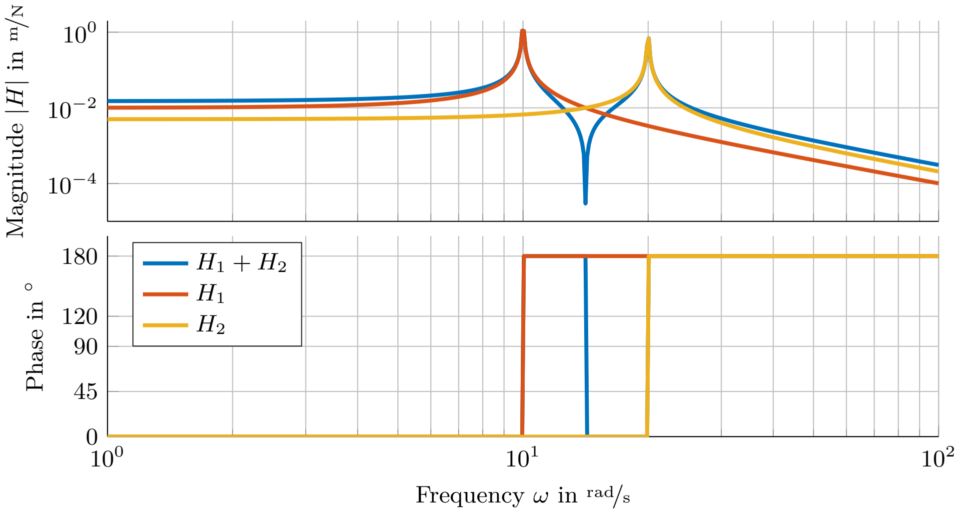


Figure 3.1: Example FRF consisting of the FRFs H_1 and H_2 corresponding to the partial transfer functions G_1 and G_2 from *Eq.* (3.34) with the parameters from *Eq.* (3.33).

The FRFs H_1 and H_2 of both uncoupled systems G_1 and G_2, respectively, as well as the sum of both FRFs are shown in *Fig.* 3.1. One can see that zero transmission occurs at the specific frequency where both uncoupled systems have the same magnitude but opposite sign, i.e. a phase difference of 180°. One can also see that H_1 and H_2 have one pole at ω_1 and ω_2, respectively, but no zeros.

3.1.3.4 Stability

A linear time-invariant dynamic system is called asymptotically stable if all poles have negative real part, i.e. $Re(s_k) < 0$. Systems with poles with negative or zero real part, i.e. $Re(s_k) \leq 0$, are called bounded-input-bounded-output-stable (BIBO-stable). If a BIBO-stable system has poles on the imaginary axis, it is undamped and called marginally stable.

Salimbahrami et al. [91] proved that the system from *Eq.* (3.1) has no poles with positive real part, and thus is BIBO-stable, if $\boldsymbol{M} \succcurlyeq 0, \boldsymbol{D} \succcurlyeq 0$, and $\boldsymbol{K} \succcurlyeq 0$, i.e. the system matrices are positive semi-definite, what is the case for matrices originating from finite element models.

3.1.4 Excitation of the Structure

3.1.4.1 Excitation Through a Dirac Pulse

The Dirac pulse (or Dirac delta function) is a fundamental signal, often used in signal analysis. It is defined as a rectangular pulse of infinitesimal width and infinite height according to

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases} \tag{3.40}$$

with an integrated weight of

$$\int_{-\infty}^{\infty} \delta(t)\, dt = 1 \; . \tag{3.41}$$

To analyse the signal spectra, the Fourier transform according to

$$u(i\omega) = \mathscr{F}\{u(t)\} = \int_{-\infty}^{\infty} u(t)\, e^{i\omega t}\, dt \tag{3.42}$$

is used. The Fourier transform for the Dirac pulse is

$$\delta(i\omega) = \mathscr{F}\{\delta(t)\} = 1 \; , \tag{3.43}$$

and thus, all frequencies are excited equally. However, Dirac pulses are not realisable in practice and can only be approximised by a pulse of small, but finite, width as introduced in *Tab.* 3.1.

3.1.4.2 Excitation Through a Unit Step

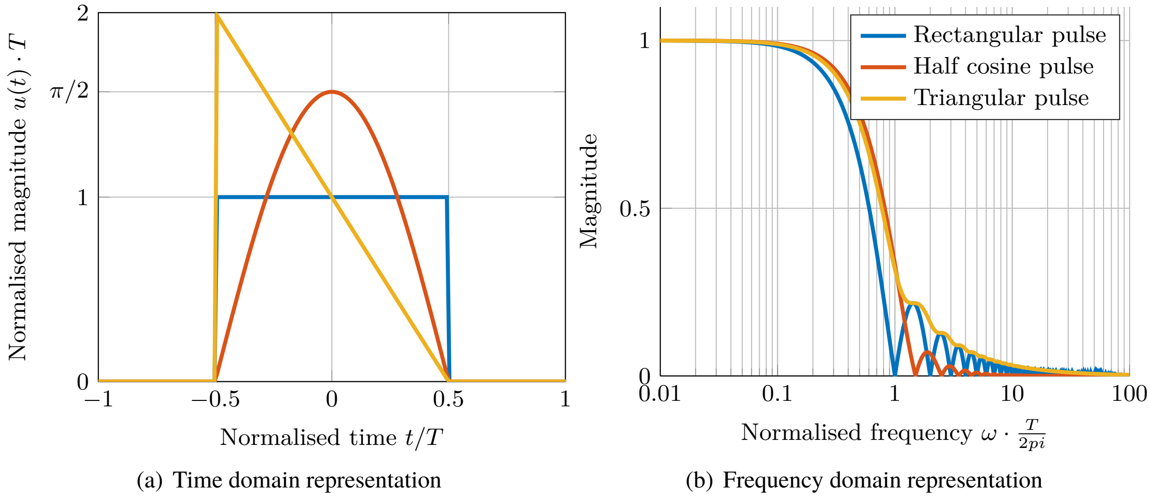
Another fundamental signal used in theory is the unit step (or Heaviside step function). It is defined as

$$\vartheta(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases} \tag{3.44}$$

and its Fourier transform is

$$\vartheta(i\omega) = \mathscr{F}\{\vartheta(t)\} = \frac{1}{i\omega} \; . \tag{3.45}$$

This shows that the magnitude of the excitation, even if theoretically spread over the whole frequency range, decays linearly with the frequency. The lower frequencies are excited most and, at some threshold, the excitation magnitude can practically be ignored. Because of the infinite length, the static gain of the spectrum, i.e. $u(i\omega)|_{\omega=0}$ is infinite. In practice, the step will be of finite length and therefore be equivalent to a rectangular pulse of large, but finite, width as shown in *Tab.* 3.1.



(a) Time domain representation (b) Frequency domain representation

Figure 3.2: Example pulse signals of finite width

3.1.4.3 Excitation Through a Pulse of Finite Width

Excitations in form of pulses of finite width arise in applications like, e.g. impact of an impact hammer, engagement of a single milling tooth, or engagement of a single grinding grain. Three example pulse shapes, a rectangular pulse, a half cosine pulse, and a triangular pulse, are shown in *Fig.* 3.2(a) and their spectral magnitudes are shown in *Fig.* 3.2(b). All of them have the same integrated value in time domain, i.e. the same area below the curves in *Fig.* 3.2(a), and thus, the same static gain in the frequency domain representation. The definitions and Fourier transforms of the signals are listed in *Tab.* 3.1.

Table 3.1: Time and frequency domain representation of different pulse types

Pulse type	$u(t) \quad \left(\text{for } -\frac{T}{2} \le t \le \frac{T}{2}\right)$	$u(i\omega) = \mathscr{F}\{u(t)\}$
Rectangular	$\frac{1}{T}$	$\frac{2\,\sin\left(\frac{T}{2}\,\omega\right)}{T\,\omega}$
Half cosine	$\frac{\pi}{2\,T}\,\cos\left(\frac{\pi}{T}\,t\right)$	$\frac{\cos\left(\frac{T}{2}\,\omega\right)}{1-\frac{T^2}{\pi^2}\,\omega^2}$
Triangular	$\frac{1}{T}\left(1-\frac{2}{T}\,t\right)$	$\frac{2}{T\,\omega}\left(\left(1-\frac{2i}{T\,\omega}\right)\,\sin\left(\frac{T}{2}\,\omega\right)+i\,\cos\left(\frac{T}{2}\,\omega\right)\right)$

Although the shape of the pulse varies for the different applications, its excitation characteristics have an important similarity. All variants show a substantial decay of magnitude near the frequency corresponding to the reciprocal value of the pulse width T, i.e. at $\omega_T = {}^{2\pi}/_{T}$. The magnitude at $10\,\omega_T$ reaches only about 3% of its static value for the rectangular and triangular pulses. The magnitude of the half cosine pulse is even lower due to its quadratic decay behaviour apparent from its Fourier transform.

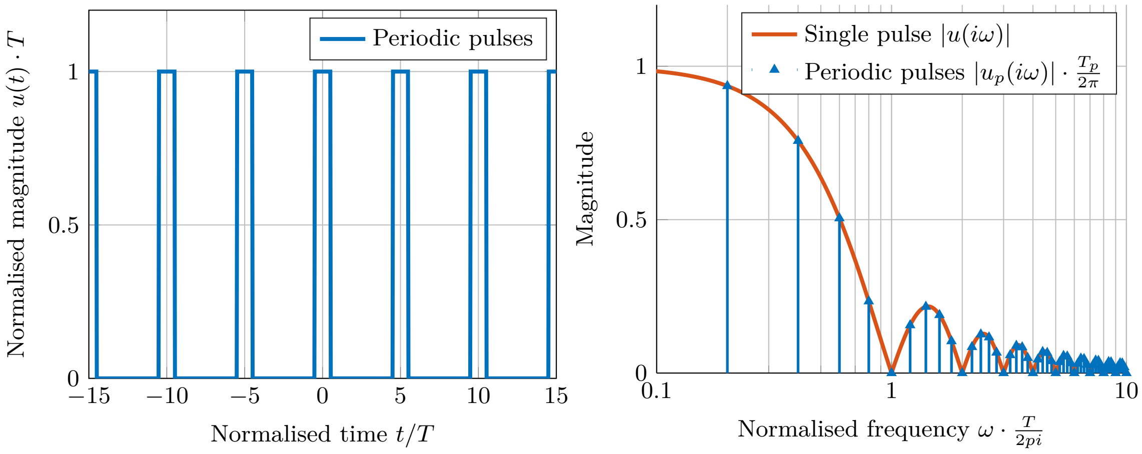
An important observation at this point is that the knowledge of the magnitude decay allows one to find a frequency limit for an arbitrary magnitude threshold.

3.1.4.4 Excitation Through Periodic Pulses

Often, periodic sequences of pulses occur, e.g. in milling processes or in gear tooth engagement. The periodic signal $u_p(t)$ can then be written as sum of arbitrary pulse signals $u(t + k\,T_p)$ shifted by the period T_p according to

$$u_p(t) = \sum_{k=-\infty}^{\infty} u(t - k\,T_p)\,, \tag{3.46}$$

as shown in *Fig.* 3.3(a) for a limited time range. The duty cycle of the signal is ${}^{T}/_{T_p}$ and describes the on-off-ratio.



(a) Time domain representation

(b) Frequency domain representation

Figure 3.3: Periodic rectangular pulses with a duty cycle of 20%

Using the Fourier transform

$$u_p(i\omega) = \mathscr{F}\left\{u_p(t)\right\} = \int_{-\infty}^{\infty} u_p(t)\,e^{-i\omega\,t}\,dt\,, \tag{3.47}$$

the signal's spectrum can be calculated according to

$$u_p(i\omega) = \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} u(t - k\,T_p)\,e^{-i\omega\,t}\,dt\,. \tag{3.48}$$

Interchanging the sum and integral leads to

$$u_p(i\omega) = \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} u(t - k\,T_p)\,e^{-i\omega\,t}\,dt \tag{3.49}$$

and the integral then corresponds to the Fourier transform of the pulse signal, shifted by $k\,T_p$, and this leads to

$$u_p(i\omega) = \sum_{k=-\infty}^{\infty} u(i\omega)\,e^{-i\omega\,k\,T_p} \tag{3.50}$$

with $e^{-i\omega\,k\,T_p}$ as the transfer function of the time delay. The constant factor $u(i\omega)$ can be factored out of the sum, leading to

$$u_p(i\omega) = u(i\omega) \sum_{k=-\infty}^{\infty} e^{-i\omega\,k\,T_p}\,. \tag{3.51}$$

The summation of complex exponentials converges to a dirac pulse train according to

$$\sum_{k=-\infty}^{\infty} e^{-i\omega\, k\, T_p} = \frac{2\pi}{T_p} \sum_{k=-\infty}^{\infty} \delta\left(\omega - k\, {}^{2\pi}\!/_{T_p}\right) \ . \tag{3.52}$$

Finally, the Fourier transform of the periodic signal reads

$$u_p(i\omega) = u(i\omega)\, \frac{2\pi}{T_p} \sum_{k=-\infty}^{\infty} \delta\left(\omega - k\, {}^{2\pi}\!/_{T_p}\right) \ . \tag{3.53}$$

This corresponds to the Fourier transform of the single pulse, sampled at all multiples of the periodic signal's fundamental frequency, and scaled by a constant factor and is strongly related to complex Fourier series and its Fourier coefficients.

It follows that the fundamental characteristic of the excitation is defined by the shape and duration of the single pulse. This is an important observation, because it shows that, once the shape of a single pulse of an excitation force is known, a threshold frequency for the periodic signal can be defined in the same manner as shown in the previous section.

In *Fig.* 3.3(*b*), the magnitude of the Fourier transform of a rectangular pulse and the magnitude of the Fourier transform of the periodic signal with a duty cycle of 20% (scaled by $T_p/2\pi$) are shown.

3.1.4.5 Excitation Through Randomly Shifted Pulses

A further extension of the idea of the superposition of single pulses is not to superpose periodically shifted pulses, but to assume a random shift for each pulse. This could represent a grinding process with randomly located grains, whereas each grain has a finite contact time on the workpiece.

An example of such a signal with superposed, randomly shifted pulses is

$$u_r(t) = \sum_{k=0}^{N} u(t - r_k\, T_r)\,, \tag{3.54}$$

whereas

$$r_k \in \mathcal{U}\,(0, T_r) \tag{3.55}$$

are uniformly distributed random numbers. Due to the linearity of the Fourier transform, the superposition of time signals can equally be represented as a superposition of their Fourier transforms as

$$u_r(i\omega) = \sum_{k=0}^{N} u(i\omega)\, e^{-i\omega\, r_k\, T_r} = u(i\omega) \sum_{k=0}^{N} e^{-i\omega\, r_k\, T_r} \tag{3.56}$$

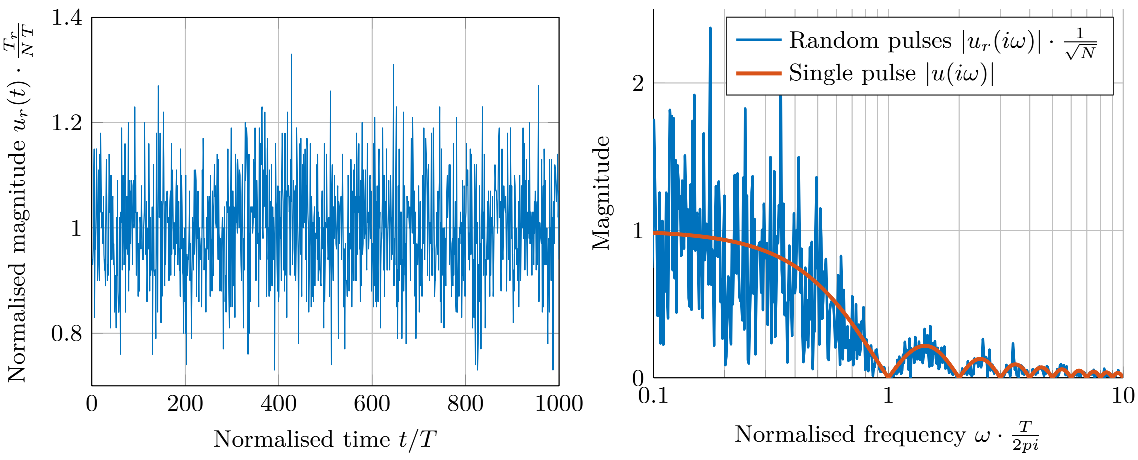
with the Fourier transform of the random time delay $e^{-i\omega\, r_k\, T_r}$.

This shows that the character of this signal, again, is dominated by the Fourier transform of the single pulse. A numeric example is shown in *Fig.* 3.4.

3.1.4.6 Excitation Through Drives and Control Loop

Beside external excitation through process forces, the machine tool structure is also excited internally by the drives and the control loop. The closed-loop FRF of a well parametrised position controller can usually be approximated as a second-order low-pass filter with a controller bandwidth ω_c and a damping ratio D as

$$H_{cl}(i\omega) = \frac{\omega_c^2}{-\omega^2 + 2\, D\, i\omega\, \omega_c + \omega_c^2} \ . \tag{3.57}$$



(a) Time domain representation

(b) Frequency domain representation

Figure 3.4: Randomly shifted rectangular pulses with $N = 100\,000$ and $\frac{T_r}{T} = 1\,000$

The high-frequency portions of the input signals will be attenuated and therefore not significantly applied to the structure. For frequencies substantially higher than the bandwidth, the transfer gain can be approximated as

$$\lim_{\omega \to \infty} |H_{cl}(i\omega)| = \frac{\omega_c}{\omega^2} \, . \tag{3.58}$$

Due to the second-order characteristic, the gain lowers quadratically with the frequency beyond the bandwidth frequency.

3.1.5 Example Systems

For the discussion and illustration of the properties of model order reduction methods, example systems are used, which are subsequently introduced.

3.1.5.1 Random Systems

Randomly generated SISO systems are used to efficiently test the properties of reduction methods on a large number of different systems with different transfer functions. The utilised systems have $N = 200$ degrees of freedom and the eigenfrequencies ω_k are uniformly distributed and sorted according to

$$\omega_k \in \mathcal{U}(500, 10^5) \, , \quad \omega_{k+1} \geq \omega_k \, , \quad k = 1, 2, \ldots, N \, . \tag{3.59}$$

The undamped systems are realised in modal coordinates with a normalised mass matrix, i.e.

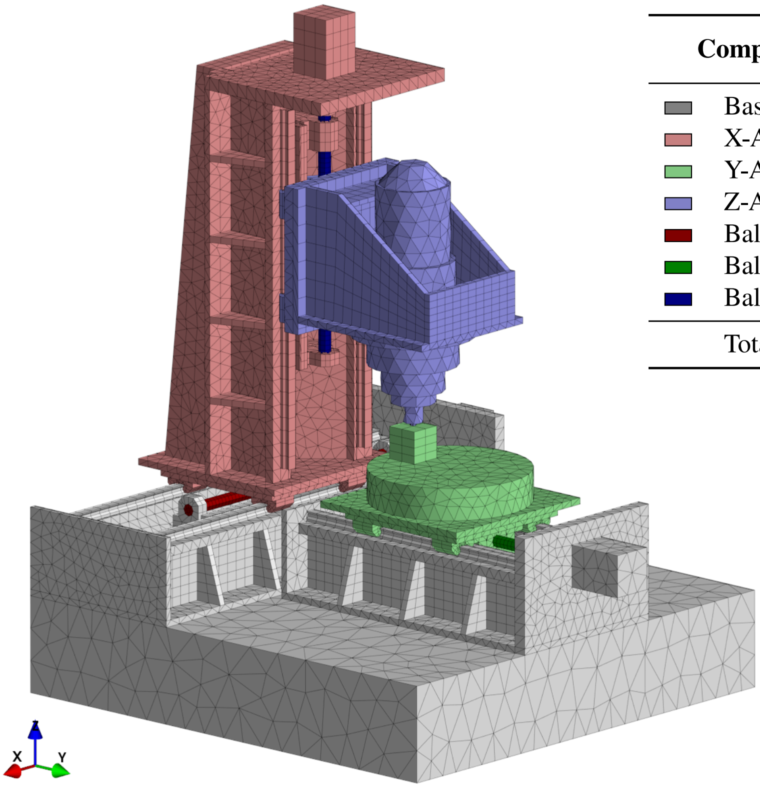
$$\boldsymbol{M} = \boldsymbol{I} \, , \quad \boldsymbol{K} = \operatorname{diag}(\omega_1^2, \omega_2^2, \ldots, \omega_N^2) \, , \tag{3.60}$$

and the input and output vectors are randomly generated as

$$\boldsymbol{b}_k \in \mathcal{U}(-1, 1) \, , \quad \boldsymbol{c}_k \in \mathcal{U}(-1, 1) \, , \quad k = 1, 2, \ldots, N \, . \tag{3.61}$$

3.1.5.2 Real-World Structure

As an example for a real structure, the reconfigurable machine tool presented by Lorenzer [70] and visualised in *Fig.* 3.5 is used. The complete model is split up into seven components, which are modelled as separate finite element models and are thus to be reduced separately. The finite element meshes consist of quadratic hexahedral, tetrahedral, and wedge elements.



Component		N
⬜	Basis	214 797
🟥	X-Axis	123 543
🟩	Y-Axis	32 244
🟦	Z-Axis	42 921
🟥	Ball screw X	11 610
🟩	Ball screw Y	8 652
🟦	Ball screw Z	9 174
	Total	442 941

Figure 3.5: Real-world example of a machine tool model with seven components, each modelled as a separate finite element model, leading to 442 941 degrees of freedom in total

3.2 Requirements on Model Order Reduction for Machine Tools

In order to find an appropriate model order reduction technique for application in machine tool simulations, it is crucial to precisely specify the requirements on such a method. Therefore, subsequently the scope of application is clarified, accuracy requirements are defined, relevant properties which should be preserved are elaborated, and restrictions on the applicable methods are listed. This finally leads to a complete set of requirements on the model order reduction method.

3.2.1 Scope of Application

The objective of the thesis is a simulation framework for the creation of machine tool models which enable in-depth analysis of the dynamics and statics of machine tools. This comprises several application fields like concept evaluation, controller design, troubleshooting, and model-based compensation or controlling tasks. Therefore, versatile analysis approaches like

- static and quasi-static analysis,
- modal analysis,
- frequency response analysis, and
- transient simulation

are particularly important and should be performable with sufficient accuracy using reduced order models.

3.2.2 Model Order Reduction Accuracy Requirements

Due to modelling simplifications and errors, the finite element models used for model order reduction do not represent the real mechanical structure exactly. A relative error for the magnitude, poles and zeros of the transfer function in the range of a few percent is nearly inevitable. Therefore, the requirement on the approximation accuracy of the reduced order model can be mitigated. As long as the reduced order model approximates the original model in the same precision scale as the original model approximates the reality, the approximation is assumed to be sufficient.

Furthermore, the transfer function of a reduced order model is not required to match the one of the original model on the whole complex plane in order to achieve accurate simulation results. It is sufficient, if the FRFs of the reduced and original model $\hat{\boldsymbol{H}}(i\omega)$ and $\boldsymbol{H}(i\omega)$, respectively, match for the frequencies which are excited by the input signal, i.e. a subset of the imaginary axis. From *Section* 3.1.4, the characteristics of the structure excitation for different excitation sources is known. All these excitation variants, excepting the Dirac pulse, which cannot be realised in practice, show a decay in magnitude for an increasing frequency. This allows to find a limit frequency ω_l above which the magnitude does not exceed a threshold value u_l as

$$|u(i\omega)| \leq u_l \quad \forall\ \omega \notin \Omega_l\ , \tag{3.62}$$

with the frequency interval of interest

$$\Omega_l = (-\omega_l,\ \omega_l)\ . \tag{3.63}$$

A threshold value between 1% and 10% of the maximum magnitude is considered appropriate in comparison with the typical modelling accuracy for finite element models. A recommendation is to choose the limit frequency ten times higher than the reciprocal value of the pulse width of the excitation or, if higher, ten times the controller bandwidth.

As error criterion for the evaluation of the model order reduction quality, the relative error for the frequency response matrix, defined as

$$\boldsymbol{Er}_{k,l}(i\omega) = \frac{\hat{\boldsymbol{H}}_{k,l}(i\omega) - \boldsymbol{H}_{k,l}(i\omega)}{\boldsymbol{H}_{k,l}(i\omega)}\ , \quad k = 1,2,\ldots,q\ , \quad l = 1,2,\ldots,p\ , \tag{3.64}$$

is used.

Inside the frequency range of interest Ω_l, the reduced order model should approximate the original model within a definable relative error bound $\bar{\varepsilon}_r$ on all FRFs as

$$\left|\boldsymbol{Er}_{k,l}(i\omega)\right| \leq \bar{\varepsilon}_r \quad \forall\ \omega \in \Omega_m - \Omega_{z_{k,l}}\ , \quad k = 1,2,\ldots,q\ , \quad l = 1,2,\ldots,p\ . \tag{3.65}$$

$\Omega_{z_{k,l}}$ is the set of frequencies in the neighbourhood of the zeros of $\boldsymbol{H}_{k,l}$, which is excluded from the relative error limitation. This mitigation of the relative error criterion is necessary, because as a consequence of the division by the original FRF $\boldsymbol{H}_{k,l}$ for the calculation of the relative error in *Eq.* (3.64), the zeros of $\boldsymbol{H}_{k,l}$ become poles of the relative error, provided that the zeros are not matched exactly. Thus, it is not practical to limit the relative error of the FRF near the zeros of the original FRF. Instead, a relative error for the places of the zeros, i.e. the zero frequencies, is defined as

$$\varepsilon_{z_k} = \frac{\hat{\omega}_{z_k} - \omega_{z_k}}{\omega_{z_k}}\ , \quad k = 1,2,\ldots,n_z \tag{3.66}$$

with the zeros of the FRFs of the original and reduced system ω_{z_k} and $\hat{\omega}_{z_k}$, respectively, and the number of zeros n_z within Ω_l. A limit for the relative zero location error $\bar{\varepsilon}_z$ is defined as

$$|\varepsilon_{z_k}| \leq \bar{\varepsilon}_z\ , \quad k = 1,2,\ldots,n_z\ . \tag{3.67}$$

The poles of the FRF are of paramount importance, because they yield to large amplification of the input signal and characterise the oscillation behaviour of the system. Moreover, the poles of the undamped system are exactly the eigenvalues of the undamped system, and thus, the result of a modal analysis. The relative error of the poles is defined as

$$\varepsilon_{p_k} = \frac{\hat{\omega}_k - \omega_k}{\omega_k} \,, \quad k = 1, 2, \ldots, n_p \tag{3.68}$$

with n_p, the number of poles in Ω_l, and the poles, i.e. the eigenfrequencies, of the original and reduced system ω_k and $\hat{\omega}_k$, respectively. The pole locations should not exceed a relative error of $\bar{\varepsilon}_p$ according to

$$|\varepsilon_{p_k}| \leq \bar{\varepsilon}_p \,, \quad k = 1, 2, \ldots, n_p \,. \tag{3.69}$$

In order to enable static analyses, the static behaviour of the reduced system is required to match the one of the original system, i.e.

$$\hat{\boldsymbol{H}}(i\omega)\big|_{\omega=0} = \boldsymbol{H}(i\omega)\big|_{\omega=0} \,. \tag{3.70}$$

With *Eq.* (3.65), (3.67), (3.69), and (3.70), the accuracy requirements on a reduced order model are defined. For the relative error limits $\bar{\varepsilon}_r$, $\bar{\varepsilon}_z$, and $\bar{\varepsilon}_p$, a value related to the overall model accuracy, between 1% and 10%, is recommended.

3.2.3 Restrictions on Model Order Reduction Methods

From the discussion of the state of the art in *Section* 2.8, some restrictions and alleviations are introduced for the model reduction method, namely:

- Projection-based model order reduction methods should be used.
- One-sided projection with equal input and output matrices has to be used for Krylov subspace methods in order to preserve stability and to achieve invariance to the realisation of the system, i.e. invariance to the choice of coordinates. This implies that all inputs to the system are to be outputs as well and vice versa.
- In order to achieve exact matching of the transfer functions of all in- and outputs at the expansion points, a block Krylov method should be used in contrast to tangential interpolation methods.
- The second-order structure of the system has to be preserved in order to be compatible with flexible multi-body simulation frameworks.
- Proportional damping has to be used in order to be able to use Krylov and eigenvector based projection bases, created using reduction methods for systems with first-order structure.
- Krylov subspace and component mode synthesis based model order reduction methods are eligible, but efficient error estimations have yet to be developed.
- For Krylov based methods, the Arnoldi algorithm should be used because of the numerical robustness.

3.3 Model Order Reduction by Krylov and Modal Subspace Projection

The key idea for the reduction method described in the subsequent sections is to combine the advantages of the modal and Krylov reduction by constructing a combined Krylov and modal subspace.

The advantage of the modal reduction is that the eigenfrequencies corresponding to the eigenvectors involved in the projection are matched exactly. The disadvantage, however, is that through the truncation of the projection basis without replacement, there are large deviations in the static and dynamic compliance, i.e. the magnitude of the FRFs, as well as in the zeros of the FRFs.

The Krylov subspace reduction has the advantage that the transfer function and its derivatives can be matched exactly at one or more discrete expansion points. On the other hand, the disadvantage is that the poles and zeros of the transfer functions are not matched explicitly and, moreover, do not converge very fast.

The hypothesis is that the combination of both methods will lead to reduced models with

- matching poles, due to the use of eigenvectors,
- matching FRFs at definable expansion points, due to the use of Krylov basis vectors,
- adequately matching FRFs in the frequency range with matched poles, and
- adequately matching zeros, because the zeros are a result of the poles and the gain of the transfer function, as explained in *Section* 3.1.3.3.

3.3.1 Modal Subspace

The realisation of a system in modal coordinates was presented in *Section* 3.1.2. The transfer function from *Eq.* (3.24) can equivalently be split into two partial sums according to

$$\boldsymbol{G}(s) = \underbrace{\sum_{k=1}^{n_{\boldsymbol{M}}} \frac{\tilde{\boldsymbol{C}}_{:k}\,\tilde{\boldsymbol{B}}_{k:}}{s^2+\omega_k^2}}_{\boldsymbol{G_M}(s)} + \underbrace{\sum_{k=n_{\boldsymbol{M}}+1}^{N} \frac{\tilde{\boldsymbol{C}}_{:k}\,\tilde{\boldsymbol{B}}_{k:}}{s^2+\omega_k^2}}_{\boldsymbol{G_R}(s)}, \tag{3.71}$$

where the first term consists of the first $n_{\boldsymbol{M}}$, and the second term of the remaining eigenvalues. The eigenvalues shall be ordered according to their absolute value increasingly. $\boldsymbol{G_M}$ can then be seen as the modally reduced model and $\boldsymbol{G_R}$ is the remainder transfer function, i.e. the absolute error of the modally reduced transfer function.

The corresponding systems are obtained by separating the modal matrix from *Eq.* (3.12) into the two parts

$$\boldsymbol{\Phi_M} = \left[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_{n_{\boldsymbol{M}}}\right] \quad \text{and} \quad \boldsymbol{\Phi_R} = \left[\boldsymbol{\phi}_{n_{\boldsymbol{M}}+1}, \boldsymbol{\phi}_{n_{\boldsymbol{M}}+2}, \ldots, \boldsymbol{\phi}_N\right] \tag{3.72}$$

and projecting the system matrices using these truncated modal matrices according to *Eq.* (3.16). $\boldsymbol{\Phi_M}$ and $\boldsymbol{\Phi_R}$ are bases for two complementary modal subspaces of the vector space $\mathbb{R}^N$.

It can be seen from *Eq.* (3.71) that the absolute error transfer function $\boldsymbol{G_R}$ does not have any poles below the eigenfrequency $\omega_{n_{\boldsymbol{M}}+1}$. This is desirable, because the poles of a transfer function do predominantly influence the transfer behaviour. However, from *Eq.* (3.26), it is clear that all modes contribute to the static behaviour of the system, and thus, the truncated system shows a static error.

3.3.2 Krylov Subspace

In *Section* 3.2.3 it has been specified that one-sided block Krylov methods are to be used with equal input and output matrices in order to preserve important properties of the system. Moreover, due to the use of proportional damping, a first-order Krylov method can be used with the undamped system in the first-order representation from *Eq.* (3.6) in order to generate Krylov subspace bases according to *Eq.* (2.32). A subspace with a projection basis $\boldsymbol{V_K}$ that complies with these conditions spans the following sum of Krylov subspaces

$$\begin{aligned}\operatorname{range}(\boldsymbol{V_K}) &= \sum_{k=1}^{n_e} \mathcal{K}_{\boldsymbol{m}_{\boldsymbol{e}_k}}\left(\left(\boldsymbol{A}-\boldsymbol{\sigma}_{\boldsymbol{e}_k}\boldsymbol{E}\right)^{-1}\boldsymbol{E},\ \left(\boldsymbol{A}-\boldsymbol{\sigma}_{\boldsymbol{e}_k}\boldsymbol{E}\right)^{-1}\boldsymbol{B}\right)\\ &= \sum_{k=1}^{n_e} \mathcal{K}_{\boldsymbol{m}_{\boldsymbol{e}_k}}\left(\left(\boldsymbol{K}+\boldsymbol{s}_{\boldsymbol{e}_k}^2\boldsymbol{M}\right)^{-1}\boldsymbol{M},\ \left(\boldsymbol{K}+\boldsymbol{s}_{\boldsymbol{e}_k}^2\boldsymbol{M}\right)^{-1}\boldsymbol{B}\right)\end{aligned} \tag{3.73}$$

with the definition of the Krylov subspace from *Eq.* (2.35), a combined input and output matrix $\boldsymbol{B} = \boldsymbol{C}^T$, a vector of n_e expansion points $\boldsymbol{\sigma_e} = -\boldsymbol{s_e}^2$, and a vector with the numbers of Krylov iterations for each expansion point $\boldsymbol{m_e}$. For $n_e = 1$, the expansion point and the number of iterations is scalar and denoted as s_e and m_e, respectively.

The sum of two subspaces $\mathscr{V}$ and $\mathscr{W}$ is defined as

$$\mathscr{V} + \mathscr{W} = \{\boldsymbol{v} + \boldsymbol{w} : \boldsymbol{v} \in \mathscr{V}, \boldsymbol{w} \in \mathscr{W}\} \ . \tag{3.74}$$

This means that all linear combinations of basis vectors of both subspaces are contained in the sum of both subspaces. The sum of two subspaces containing one line each, e.g., is the plane spanned by those two lines.

By selecting an expansion point at zero, i.e. $0 \in \boldsymbol{s_e}$, the static stiffness of the original system is matched exactly by the reduced system (within the numerical precision). However, in contrast to modal reduction, there is no guarantee that any poles are matched exactly.

3.3.3 Krylov and Modal Subspace

The Krylov and modal subspace (KMS) is the sum of a Krylov subspace and a modal subspace. The sum of the Krylov and modal subspaces spanned by the bases $\boldsymbol{V_K}$ and $\boldsymbol{\Phi_M}$, respectively, is

$$\text{range}(\boldsymbol{V}) = \text{range}(\boldsymbol{V_K}) + \text{range}(\boldsymbol{\Phi_M}) \tag{3.75}$$

with the KMS basis matrix $\boldsymbol{V}$, which can be obtained through orthogonalisation of the combination of both basis matrices according to

$$\boldsymbol{V} = \text{orth}\left(\begin{bmatrix}\boldsymbol{V_K} & \boldsymbol{\Phi_M}\end{bmatrix}\right) \ , \tag{3.76}$$

where the rank of the new basis matrix, and thus the dimension of the combined subspace, can be smaller than the sum of the dimensions of both subspaces, i.e.

$$\text{rank}(\boldsymbol{V}) \leq \text{rank}(\boldsymbol{V_K}) + \text{rank}(\boldsymbol{\Phi_M}) \, , \tag{3.77}$$

because $\boldsymbol{V_K}$ and $\boldsymbol{\Phi_M}$ do potentially contain linear dependent basis vectors.

Because, as shown in *Eq.* (2.19), the residuum of the system is orthogonal to the subspace used for projection and does not depend on the chosen basis for the subspace, the combination of subspaces leads to residua which are orthogonal to the combined subspace. Thus, the desired properties of matching poles and matching moments of the transfer function are combined with the KMS method.

3.3.4 Comparison of the Methods by Means of an Example

For the illustration of the properties of modal subspace, Krylov subspace, and KMS reduction, the methods are applied to a random system as described in *Section* 3.1.5.1 with the FRF shown in *Fig.* 3.6. The combined input matrix for the SISO example system consists of the input and output vectors according to

$$\boldsymbol{B} = \begin{bmatrix}\boldsymbol{b} & \boldsymbol{c}^T\end{bmatrix} \ , \quad \boldsymbol{B} \in \mathbb{R}^{N \times 2} \ . \tag{3.78}$$

The Krylov subspace is expanded at $s_e = 0$ in $m_e = 5$ iterations, which leads to $n_K = 10$ degrees of freedom due to the two vectors in the matrix $\boldsymbol{B}$. The modal subspace is created using $n_M = 10$ eigenvectors, and the KMS is the combination of two Krylov vectors ($m_e = 1$, $s_e = 0$) and eight eigenvectors, also leading to $n = 10$ degrees of freedom.

Fig. 3.7 shows the absolute error of the FRFs. As explained before, the modal subspace reduction matches the poles exactly, and thus, the absolute error has no poles or zeros in the range of the first ten

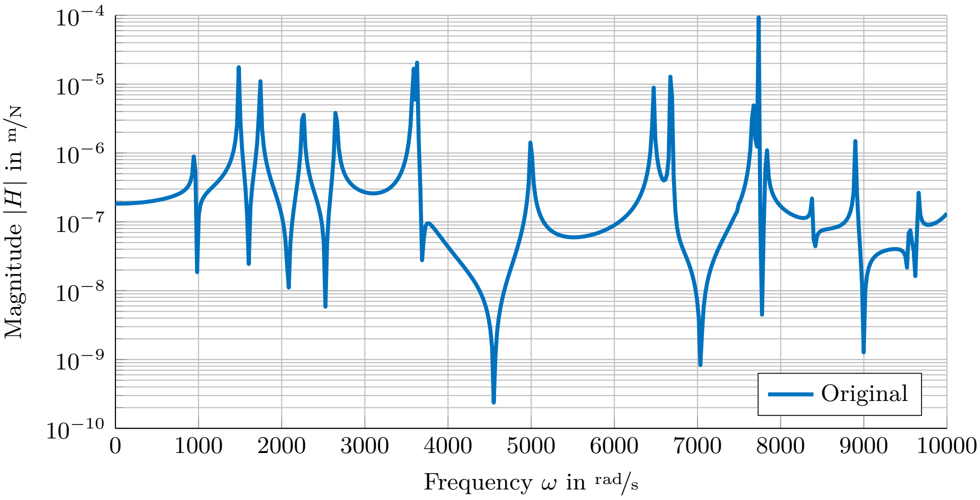


Figure 3.6: FRF of a randomly generated example system

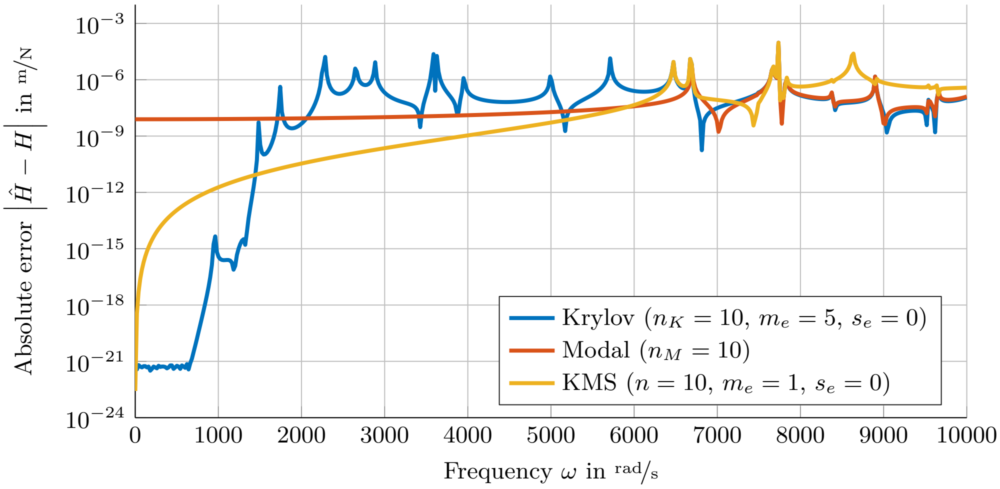


Figure 3.7: Absolute transfer function error for models reduced using Krylov subspace, modal subspace, and KMS

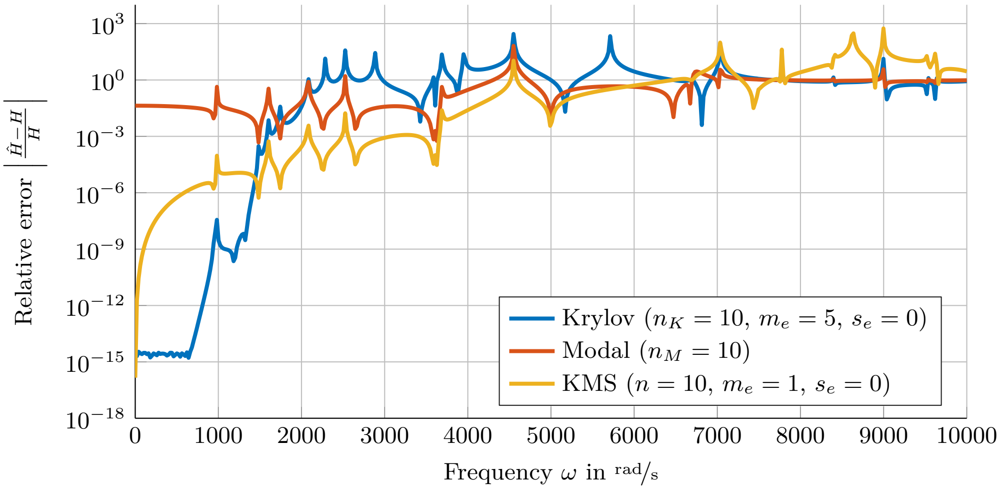


Figure 3.8: Relative transfer function error for models reduced using Krylov subspace, modal subspace, and KMS

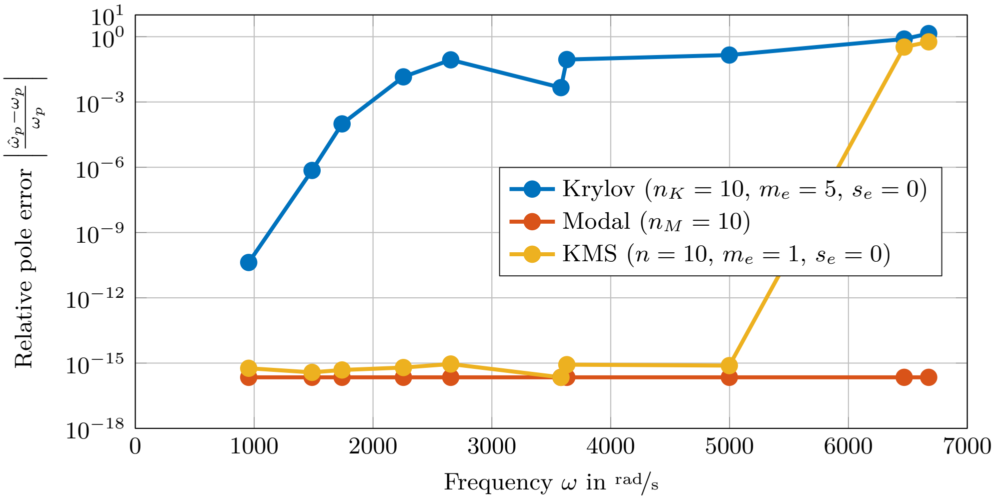


Figure 3.9: Relative pole frequency error for the first 10 poles for models reduced using Krylov subspace, modal subspace, and KMS

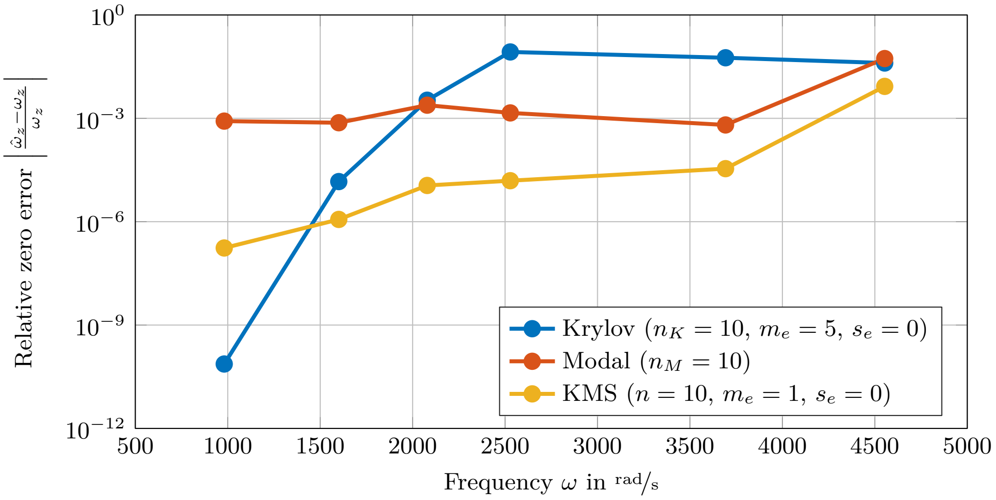


Figure 3.10: Relative zero frequency error for the first 6 zeros for models reduced using Krylov subspace, modal subspace, and KMS

Table 3.2: Maximum frequency for which the accuracy criteria from *Section* 3.2.2 are fulfilled for the example system. ($\bar\varepsilon_r = \bar\varepsilon_z = \bar\varepsilon_p = 5\%$)

	Relative transfer error limit frequency	**Relative zero error limit frequency**	**Relative pole error limit frequency**
	max $\omega : \left\|\frac{\hat H(i\omega)-H(i\omega)}{H(i\omega)}\right\| \le \bar\varepsilon_r$	max $\omega_z : \left\|\frac{\hat\omega_z-\omega_z}{\omega_z}\right\| \le \bar\varepsilon_z$	max $\omega_p : \left\|\frac{\hat\omega_p-\omega_p}{\omega_p}\right\| \le \bar\varepsilon_p$
Krylov	2 010 rad/s	2 000 rad/s	2 000 rad/s
Modal	3 670 rad/s	3 690 rad/s	3 670 rad/s
KMS	4 130 rad/s	4 550 rad/s	4 130 rad/s

modes of the original system. The Krylov subspace reduction leads to numerically exact matching of the FRF at the expansion point $s_0 = 0$. For increasing frequencies, the approximation is rapidly losing quality. The poles are not matched exactly, and thus, the error of the FRF grows rapidly during the first couple of poles. The reduction using KMS leads to both, an exactly matched static gain as well as exactly matched poles, and thus, a good approximation quality within the frequency range of the first eight poles. The matching accuracy for the poles is shown in *Fig.* 3.9.

The relative error shown in *Fig.* 3.8 has poles and zeros for all methods. As explained in *Section* 3.2.2, these are introduced through the division by the original FRF in the calculation of the relative error. When ignoring the frequencies near the zeros of the original transfer function, the KMS method has the broadest range of good approximation.

The relative error for the zeros of the reduced FRFs are shown in *Fig.* 3.10 for all three methods. Apart from the first zero, the KMS method matches the zeros of the original transfer function best.

Overall, for the example system, the combined Krylov and modal subspace reduction clearly outmatches both the Krylov and modal reduction methods in terms of approximation quality. The maximum frequencies in which the requirements defined in *Section* 3.2.2 are fulfilled, are summarised in *Tab.* 3.2. In this example, the KMS reduction enlarges the usable frequency range by more than factor two in comparison with Krylov reduction. The modal reduction shows decent results in this table, but is inapplicable because the static stiffness is not matched accurately.

The comparison of the KMS method with pure modal and Krylov subspace reduction shows the potential of the KMS method. However, the relative error of the FRF as shown in *Fig.* 3.8 as well as the relative error of the zeros as shown in *Fig.* 3.10 have to be further characterised in order to meet the accuracy requirements from *Section* 3.2.2. In the next section, the method is further developed and an error estimation is derived, which then yields to a straightforward selection of reduction parameters.

3.4 Error Estimation for KMS Reduction

3.4.1 Separation of the Relative Error

Here, the idea is to separate the relative error for KMS based reduction into two factors, the relative error for modal reduction and the relative error for Krylov reduction of the remainder system after modal reduction, and then estimate those two factors separately. In order to proof this separation, some important properties have to be proven.

First, it is shown that a KMS created using a Krylov subspace of the original system is equal to the KMS created using a Krylov subspace of the remainder system, i.e. the system which is reduced by the truncated modal matrix $\boldsymbol{\Phi_R}$.

Theorem 3.4.1. *Let* $\boldsymbol{\Phi_M}$ *and* $\boldsymbol{\Phi_R}$ *be two modal bases according to Eq.* (3.72)*,* $\boldsymbol{V_K}$ *a Krylov basis generated from the original system,* $\boldsymbol{V_{RK}}$ *a Krylov basis generated from the residual system which is the original system projected by* $\boldsymbol{\Phi_R}$*.*

The KMS created using the modal basis $\boldsymbol{\Phi_M}$ *and the Krylov basis* $\boldsymbol{V_K}$ *is equal to the KMS created using the modal basis* $\boldsymbol{\Phi_M}$ *and the Krylov basis* $\boldsymbol{V_{RK}}$*, i.e.*

$$\mathit{range}\,(\boldsymbol{V_K}) + \mathit{range}\,(\boldsymbol{\Phi_M}) = \mathit{range}\,(\boldsymbol{V_{RK}}) + \mathit{range}\,(\boldsymbol{\Phi_M}) \tag{3.79}$$

Proof. The transformation of the original system matrices into modal coordinates can be written as

$$\begin{aligned}
\tilde{\boldsymbol{K}} &= \boldsymbol{\Phi}^T \boldsymbol{K} \boldsymbol{\Phi} = \operatorname{diag}\left(\omega_1^2, \omega_2^2, \ldots, \omega_N^2\right) = \boldsymbol{\Lambda} \\
\tilde{\boldsymbol{M}} &= \boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{\Phi} = \boldsymbol{I} \\
\tilde{\boldsymbol{b}} &= \boldsymbol{\Phi}^T \boldsymbol{b}\ .
\end{aligned} \tag{3.80}$$

Because of the diagonal structure, the stiffness matrix can be split into two decoupled matrices $\boldsymbol{\Lambda_M}$ and $\boldsymbol{\Lambda_R}$ for the two complementary modal bases $\boldsymbol{\Phi_M}$ and $\boldsymbol{\Phi_R}$, respectively, as

$$\boldsymbol{\Lambda} = \boldsymbol{\Phi}^T \boldsymbol{K} \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi_M} & \boldsymbol{\Phi_R} \end{bmatrix}^T \boldsymbol{K} \begin{bmatrix} \boldsymbol{\Phi_M} & \boldsymbol{\Phi_R} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda_M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda_R} \end{bmatrix} . \tag{3.81}$$

The Krylov subspace which is spanned by the basis $\boldsymbol{V_K}$ is defined as

$$\mathrm{range}\left(\boldsymbol{V_K}\right) = \mathscr{K}_{m_e}\left(\left(\boldsymbol{K} + s_e^2 \boldsymbol{M}\right)^{-1} \boldsymbol{M},\ \left(\boldsymbol{K} + s_e^2 \boldsymbol{M}\right)^{-1} \boldsymbol{b}\right) . \tag{3.82}$$

The starting vector can be written as

$$\left(\boldsymbol{K} + s_e^2 \boldsymbol{M}\right)^{-1} \boldsymbol{b} = \boldsymbol{\Phi}\boldsymbol{\Gamma}\tilde{\boldsymbol{b}} = \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T \boldsymbol{b} = \boldsymbol{v}_{\boldsymbol{K}_0} \tag{3.83}$$

with

$$\boldsymbol{\Gamma} = \left(\boldsymbol{K} + s_e^2 \boldsymbol{M}\right)^{-1} . \tag{3.84}$$

Note that the coordinate transformation from modal coordinates to original coordinates

$$\boldsymbol{x} = \boldsymbol{\Phi}\tilde{\boldsymbol{x}} \tag{3.85}$$

is used here. By splitting $\boldsymbol{\Gamma}$ into its decoupled parts

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma_M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Gamma_R} \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{\Lambda_M} + s_0^2 \boldsymbol{I}\right)^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \left(\boldsymbol{\Lambda_R} + s_0^2 \boldsymbol{I}\right)^{-1} \end{bmatrix} , \tag{3.86}$$

Eq. (3.83) can be separated into two parts according to

$$\boldsymbol{v}_{\boldsymbol{K}_0} = \begin{bmatrix} \boldsymbol{\Phi_M} & \boldsymbol{\Phi_R} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Gamma_M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Gamma_R} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi_M} & \boldsymbol{\Phi_R} \end{bmatrix}^T \boldsymbol{b} = \underbrace{\boldsymbol{\Phi_M}\boldsymbol{\Gamma_M}\boldsymbol{\Phi}_{\boldsymbol{M}}^T \boldsymbol{b}}_{\boldsymbol{v}_{\boldsymbol{M}_0}} + \underbrace{\boldsymbol{\Phi_R}\boldsymbol{\Gamma_R}\boldsymbol{\Phi}_{\boldsymbol{R}}^T \boldsymbol{b}}_{\boldsymbol{v}_{\boldsymbol{R}_0}} , \tag{3.87}$$

leading to starting vectors for the Krylov subspaces for the modally reduced and residual system, $\boldsymbol{v}_{\boldsymbol{M}_0}$ and $\boldsymbol{v}_{\boldsymbol{R}_0}$, respectively.

Similarly, the Krylov recursion for the calculation of further basis vectors can be written in modal coordinates according to

$$\boldsymbol{v}_{\boldsymbol{K}_k} = \left(\boldsymbol{K} + s_0^2 \boldsymbol{M}\right)^{-1} \boldsymbol{M}\, \boldsymbol{v}_{\boldsymbol{K}_{k-1}} = \boldsymbol{\Phi}\,\boldsymbol{\Gamma} \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{v}_{\boldsymbol{K}_{k-1}} \tag{3.88}$$

and separated into two parts as

$$\boldsymbol{v}_{\boldsymbol{K}_k} = \underbrace{\boldsymbol{\Phi_M}\boldsymbol{\Gamma_M} \left(\boldsymbol{\Phi}_{\boldsymbol{M}}^T \boldsymbol{\Phi_M}\right)^{-1} \boldsymbol{\Phi}_{\boldsymbol{M}}^T \boldsymbol{v}_{\boldsymbol{M}_{k-1}}}_{\boldsymbol{v}_{\boldsymbol{M}_k} \in \mathrm{range}(\boldsymbol{\Phi_M})} + \underbrace{\boldsymbol{\Phi_R}\boldsymbol{\Gamma_R} \left(\boldsymbol{\Phi}_{\boldsymbol{R}}^T \boldsymbol{\Phi_R}\right)^{-1} \boldsymbol{\Phi}_{\boldsymbol{R}}^T \boldsymbol{v}_{\boldsymbol{R}_{k-1}}}_{\boldsymbol{v}_{\boldsymbol{R}_k} \in \mathrm{range}(\boldsymbol{\Phi_R})} . \tag{3.89}$$

Note that here the previous basis vector $\boldsymbol{v}_{\boldsymbol{K}_{k-1}}$ is transformed from original coordinates to modal coordinates using the transformation

$$\tilde{\boldsymbol{x}} = \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{x} . \tag{3.90}$$

Finally, the basis vectors for the Krylov subspaces created from the different system parts are put together to the basis matrices

$$\begin{aligned} \boldsymbol{V_K} &= \begin{bmatrix} \boldsymbol{v}_{\boldsymbol{K}_0} & \boldsymbol{v}_{\boldsymbol{K}_1} & \dots & \boldsymbol{v}_{\boldsymbol{K}_{me-1}} \end{bmatrix} \\ \boldsymbol{V_M} &= \begin{bmatrix} \boldsymbol{v}_{\boldsymbol{M}_0} & \boldsymbol{v}_{\boldsymbol{M}_1} & \dots & \boldsymbol{v}_{\boldsymbol{M}_{me-1}} \end{bmatrix} \\ \boldsymbol{V_{RK}} &= \begin{bmatrix} \boldsymbol{v}_{\boldsymbol{R}_0} & \boldsymbol{v}_{\boldsymbol{R}_1} & \dots & \boldsymbol{v}_{\boldsymbol{R}_{me-1}} \end{bmatrix} \ . \end{aligned} \tag{3.91}$$

From *Eq.* (3.87) and *Eq.* (3.89) it can be concluded that

$$\boldsymbol{V_K} = \boldsymbol{V_M} + \boldsymbol{V_{RK}} \ . \tag{3.92}$$

The KMS is defined in *Eq.* (3.75) as

$$\operatorname{range}(\boldsymbol{V}) = \operatorname{range}(\boldsymbol{V_K}) + \operatorname{range}(\boldsymbol{\Phi_M}) \ , \tag{3.93}$$

which shows that the modal subspace spanned by $\boldsymbol{\Phi_M}$ is explicitly added to the Krylov subspace. Therefore, the Krylov basis can be projected orthogonally to the modal subspace, leading to the same KMS result. The projection according to *Eq.* (2.18) for projection into the complementary subspace to range $(\boldsymbol{\Phi_M})$ is

$$\bar{\boldsymbol{P}}_{\boldsymbol{M}} = \boldsymbol{I} - \boldsymbol{\Phi_M} \left(\boldsymbol{\Phi_M}^T \boldsymbol{\Phi_M} \right)^{-1} \boldsymbol{\Phi_M}^T \ . \tag{3.94}$$

Using *Eq.* (3.92), the Krylov basis can be projected according to

$$\bar{\boldsymbol{P}}_{\boldsymbol{M}} \boldsymbol{V_K} = \bar{\boldsymbol{P}}_{\boldsymbol{M}} \left(\boldsymbol{V_{RK}} + \boldsymbol{V_M} \right) = \bar{\boldsymbol{P}}_{\boldsymbol{M}} \boldsymbol{V_{RK}} + \underbrace{\bar{\boldsymbol{P}}_{\boldsymbol{M}} \boldsymbol{V_M}}_{\boldsymbol{0}} = \bar{\boldsymbol{P}}_{\boldsymbol{M}} \boldsymbol{V_{RK}} \ . \tag{3.95}$$

Because the basis $\boldsymbol{V_M}$ lies in the modal subspace spanned by $\boldsymbol{\Phi_M}$, the term $\bar{\boldsymbol{P}}_{\boldsymbol{M}} \boldsymbol{V_M}$ vanishes and the projection of the Krylov bases created with both the original and the residual system are equal, if projected with $\bar{\boldsymbol{P}}_{\boldsymbol{M}}$. It follows for the KMS

$$\begin{aligned} \operatorname{range}(\boldsymbol{V}) &= \operatorname{range}(\boldsymbol{V_K}) + \operatorname{range}(\boldsymbol{\Phi_M}) \\ &= \operatorname{range}\left(\bar{\boldsymbol{P}}_{\boldsymbol{M}} \boldsymbol{V_K}\right) + \operatorname{range}(\boldsymbol{\Phi_M}) \\ &= \operatorname{range}\left(\bar{\boldsymbol{P}}_{\boldsymbol{M}} \boldsymbol{V_{RK}}\right) + \operatorname{range}(\boldsymbol{\Phi_M}) \\ &= \operatorname{range}(\boldsymbol{V_{RK}}) + \operatorname{range}(\boldsymbol{\Phi_M}) \ . \end{aligned} \tag{3.96}$$

□

As Freund [43] already showed, the transfer function of a reduced system is invariant to the chosen basis as long as it spans the same subspace.

Lemma 3.4.1. *The transfer function of a system reduced by projection is independent of the particular choice of the projection basis for the same subsystem.*

Proof. Let $\boldsymbol{V} \in \mathbb{R}^{N \times n}$ be a basis for the subspace $\mathscr{V} = \operatorname{range}(V)$. Let $\boldsymbol{L} \in \mathbb{R}^{N \times N}$ be a non-singular matrix. Then $\bar{\boldsymbol{V}} = \boldsymbol{L}\boldsymbol{V}$ is also a basis for the subspace $\mathscr{V}$. One-sided projection of the system matrices by $\boldsymbol{V}$ leads to

$$\hat{\boldsymbol{M}} = \boldsymbol{V}^T \boldsymbol{M} \boldsymbol{V} \ , \qquad \hat{\boldsymbol{D}} = \boldsymbol{V}^T \boldsymbol{D} \boldsymbol{V} \ , \qquad \hat{\boldsymbol{K}} = \boldsymbol{V}^T \boldsymbol{K} \boldsymbol{V} \ , \tag{3.97}$$

$$\hat{\boldsymbol{B}} = \boldsymbol{V}^T \boldsymbol{B} \ , \qquad \hat{\boldsymbol{C}} = \boldsymbol{C} \boldsymbol{V} \tag{3.98}$$

with the transfer function

$$\hat{\boldsymbol{G}}(s) = \hat{\boldsymbol{C}} \left(\hat{\boldsymbol{M}} s^2 + \hat{\boldsymbol{D}} s + \hat{\boldsymbol{K}} \right)^{-1} \hat{\boldsymbol{B}} \ . \tag{3.99}$$

A projection of the system matrices by $\bar{\boldsymbol{V}}$, expressed by the projected matrices from *Eq.* (3.98) using the equality $\bar{\boldsymbol{V}} = \boldsymbol{L}\boldsymbol{V}$ leads to

$$\bar{\boldsymbol{M}} = \boldsymbol{L}^T \hat{\boldsymbol{M}} \boldsymbol{L}\,, \qquad \bar{\boldsymbol{D}} = \boldsymbol{L}^T \hat{\boldsymbol{D}} \boldsymbol{L}\,, \qquad \bar{\boldsymbol{K}} = \boldsymbol{L}^T \hat{\boldsymbol{K}} \boldsymbol{L}\,, \tag{3.100}$$

$$\bar{\boldsymbol{B}} = \boldsymbol{L}^T \hat{\boldsymbol{B}}\,, \qquad \bar{\boldsymbol{C}} = \hat{\boldsymbol{C}} \boldsymbol{L} \tag{3.101}$$

and the transfer function

$$\begin{aligned}\bar{\boldsymbol{G}}(s) &= \hat{\boldsymbol{C}}\boldsymbol{L}\,\left(\boldsymbol{L}^T\hat{\boldsymbol{M}}\boldsymbol{L}\,s^2 + \boldsymbol{L}^T\hat{\boldsymbol{D}}\boldsymbol{L}\,s + \boldsymbol{L}^T\hat{\boldsymbol{K}}\boldsymbol{L}\right)^{-1}\,\boldsymbol{L}^T\hat{\boldsymbol{B}}\\ &= \hat{\boldsymbol{C}}\boldsymbol{L}\,\left(\boldsymbol{L}^T\left(\hat{\boldsymbol{M}}\,s^2 + \hat{\boldsymbol{D}}\,s + \hat{\boldsymbol{K}}\right)\boldsymbol{L}\right)^{-1}\,\boldsymbol{L}^T\hat{\boldsymbol{B}}\\ &= \hat{\boldsymbol{C}}\,\left(\hat{\boldsymbol{M}}\,s^2 + \hat{\boldsymbol{D}}\,s + \hat{\boldsymbol{K}}\right)^{-1}\,\hat{\boldsymbol{B}} = \hat{\boldsymbol{G}}(s)\,,\end{aligned} \tag{3.102}$$

which is equal to the transfer function of *Eq.* (3.99). □

Using *Theorem* 3.4.1 and *Lemma* 3.4.1, it can be shown that the FRF of the KMS reduced system can be split into a sum of the FRF of the modally reduced system and the one of the Krylov reduction of the remainder system.

Corollary 3.4.1. *Let* $\boldsymbol{H}$ *be the FRF of the original system split into two parts,* $\boldsymbol{H_M}$ *and* $\boldsymbol{H_R}$*, i.e. the FRFs of the systems reduced by* $\boldsymbol{\Phi_M}$ *and* $\boldsymbol{\Phi_R}$ *from Eq.* (3.72)*, respectively, such that*

$$\boldsymbol{H} = \boldsymbol{H_M} + \boldsymbol{H_R}\ . \tag{3.103}$$

Let $\boldsymbol{H_{RK}}$ *be the FRF of the system reduced by* $\boldsymbol{V_{RK}}$ *from Theorem* 3.4.1*, and* $\boldsymbol{H_E}$ *the corresponding error transfer function, such that*

$$\boldsymbol{H_R} = \boldsymbol{H_{RK}} + \boldsymbol{H_E}\ . \tag{3.104}$$

Let $\hat{\boldsymbol{H}}$ *be the FRF of the system reduced by KMS reduction with the projection basis* **V** *from Eq.* (3.77)*. Then* $\hat{\boldsymbol{H}}$ *can be expressed as sum of the FRFs* $\boldsymbol{H_M}$ *and* $\boldsymbol{H_{RK}}$ *according to*

$$\hat{\boldsymbol{H}} = \boldsymbol{H_M} + \boldsymbol{H_{RK}}\ . \tag{3.105}$$

Proof. Inserting *Eq.* (3.104) into *Eq.* (3.103) leads to

$$\boldsymbol{H} = \boldsymbol{H_M} + \boldsymbol{H_R} = \boldsymbol{H_M} + \boldsymbol{H_{RK}} + \boldsymbol{H_E}\ . \tag{3.106}$$

As shown in *Theorem* 3.4.1 the KMS generated by $\boldsymbol{V_K}$ and $\boldsymbol{\Phi_M}$ spans the same subspace as the KMS from $\boldsymbol{V_{RK}}$ and $\boldsymbol{\Phi_M}$. Furthermore, as shown in *Lemma* 3.4.1, the resulting transfer function depends only on the subspace and not on the particular basis of the subspace, and thus, the error of $\hat{\boldsymbol{H}}$ has to be equal to $\boldsymbol{H_E}$, i.e.

$$\boldsymbol{H} = \hat{\boldsymbol{H}} + \boldsymbol{H_E}\ . \tag{3.107}$$

Combining *Eq.* (3.106) and *Eq.* (3.107) leads to

$$\hat{\boldsymbol{H}} = \boldsymbol{H_M} + \boldsymbol{H_{RK}}\ . \tag{3.108}$$

□

This allows to show that the relative error of the KMS-reduced system can be represented as product of the relative error of the modally reduced system and the relative error of the remainder system reduced using Krylov subspace reduction.

Theorem 3.4.2. *Let $\boldsymbol{H}$ be the FRF of the original system separated into two parts, $\boldsymbol{H_M}$, the FRF of the system reduced by $\boldsymbol{\Phi_M}$ and $\boldsymbol{H_R}$, the FRF of the system reduced by $\boldsymbol{\Phi_R}$, such that*

$$\boldsymbol{H} = \boldsymbol{H_M} + \boldsymbol{H_R} \; . \tag{3.109}$$

Let $\boldsymbol{H_{RK}}$ be the FRF of the Krylov reduction of $\boldsymbol{H_R}$ and $\boldsymbol{H_E}$ the error of this reduction according to

$$\boldsymbol{H_R} = \boldsymbol{H_{RK}} + \boldsymbol{H_E} \; . \tag{3.110}$$

Then the relative error of the FRF of the KMS reduced system is the negative product of the relative error from modal reduction and the relative error of the Krylov reduction of the remainder system $\boldsymbol{H_R}$ according to

$$\frac{\hat{\boldsymbol{H}}_{k,l} - \boldsymbol{H}_{k,l}}{\boldsymbol{H}_{k,l}} = -\frac{\boldsymbol{H_M}_{k,l} - \boldsymbol{H}_{k,l}}{\boldsymbol{H}_{k,l}} \frac{\boldsymbol{H_{RK}}_{k,l} - \boldsymbol{H_R}_{k,l}}{\boldsymbol{H_R}_{k,l}} \, , \quad k = 1, 2, \ldots, q \, , \quad l = 1, 2, \ldots, p \, . \tag{3.111}$$

Proof. The second factor of *Eq.* (3.111) can, by replacing $\boldsymbol{H_R} = \boldsymbol{H} - \boldsymbol{H_M}$, be written as

$$\frac{\boldsymbol{H_{RK}}_{k,l} - \boldsymbol{H_R}_{k,l}}{\boldsymbol{H_R}_{k,l}} = \frac{\boldsymbol{H_{RK}}_{k,l} - \left(\boldsymbol{H}_{k,l} - \boldsymbol{H_M}_{k,l}\right)}{\boldsymbol{H}_{k,l} - \boldsymbol{H_M}_{k,l}} = -\frac{\boldsymbol{H_{RK}}_{k,l} + \boldsymbol{H_M}_{k,l} - \boldsymbol{H}_{k,l}}{\boldsymbol{H_M}_{k,l} - \boldsymbol{H}_{k,l}} \; . \tag{3.112}$$

From *Corollary* 3.4.1, the term $\boldsymbol{H_M} + \boldsymbol{H_{RK}}$ can be replaced as

$$\boldsymbol{H_M} + \boldsymbol{H_{RK}} = \hat{\boldsymbol{H}} \tag{3.113}$$

which leads to

$$\frac{\boldsymbol{H_{RK}}_{k,l} - \boldsymbol{H_R}_{k,l}}{\boldsymbol{H_R}_{k,l}} = -\frac{\hat{\boldsymbol{H}}_{k,l} - \boldsymbol{H}_{k,l}}{\boldsymbol{H_M}_{k,l} - \boldsymbol{H}_{k,l}} \; . \tag{3.114}$$

Reinserting this as the second factor of the right-hand side of *Eq.* (3.111) leads to

$$-\frac{\boldsymbol{H_M}_{k,l} - \boldsymbol{H}_{k,l}}{\boldsymbol{H}_{k,l}} \frac{\boldsymbol{H_{RK}}_{k,l} - \boldsymbol{H_R}_{k,l}}{\boldsymbol{H_R}_{k,l}} = -\frac{\hat{\boldsymbol{H}}_{k,l} - \boldsymbol{H}_{k,l}}{\boldsymbol{H_M}_{k,l} - \boldsymbol{H}_{k,l}} \frac{\boldsymbol{H_M}_{k,l} - \boldsymbol{H}_{k,l}}{\boldsymbol{H}_{k,l}} = \frac{\hat{\boldsymbol{H}}_{k,l} - \boldsymbol{H}_{k,l}}{\boldsymbol{H}_{k,l}} \; . \tag{3.115}$$

□

The factor

$$\boldsymbol{E_M}_{k,l} = \frac{\boldsymbol{H_M}_{k,l} - \boldsymbol{H}_{k,l}}{\boldsymbol{H}_{k,l}} \tag{3.116}$$

is the relative error of the modally reduced system related to the original system and the factor

$$\boldsymbol{E_{RK}}_{k,l} = \frac{\boldsymbol{H_{RK}}_{k,l} - \boldsymbol{H_R}_{k,l}}{\boldsymbol{H_R}_{k,l}} \tag{3.117}$$

is the relative error of the Krylov subspace based reduction of the remainder system, related to the remainder system. Subsequently, both error factors will be analysed in order to find an estimation for the total relative error.

3.4.2 Estimation of the Relative Error for Krylov Subspace Based Reduction

The fact that the FRF of the remainder system $\boldsymbol{H_R}$ does not have any poles in the frequency range below the first neglected eigenfrequency of the modal reduction, as explained in *Section* 3.3.1, allows restricting the estimation of the relative error of Krylov subspace reduction to the frequency range below the first pole.

3.4.2.1 Hypothesis for the Error Bound

Krylov subspace reduction leads to Padé approximation of the transfer function from the starting vector, i.e. the input or output vector, to the state vector. Thus, the eigenvalues nearest to the expansion point s_e which are controllable or observable tend to be matched accurately. If these eigenvalues, however, are only controllable or observable but not both, they won't affect the transfer function. This is utilised here to construct a test system which leads to poor approximation in order to find an upper bound estimation for the relative error.

The test system exhibits two eigenvalues at ω_0 and one at $\eta\omega_0$ with $\eta \geq 0$ and is realised in modal coordinates according to

$$\boldsymbol{M} = \boldsymbol{I} \ , \quad \boldsymbol{K} = \operatorname{diag}\left(\omega_0^2, \omega_0^2, (\eta\omega_0)^2\right) \ , \quad \boldsymbol{b} = \begin{bmatrix}1 & 0 & 1\end{bmatrix}^T \ , \quad \boldsymbol{c} = \begin{bmatrix}0 & 1 & 1\end{bmatrix} \ . \tag{3.118}$$

The input vector $\boldsymbol{b}$ affects the first and the third state, the output vector $\boldsymbol{c}$ the second and third state. The first and second states are thus controllable or observable but not both and do, therefore, not influence the transfer function, which reads

$$G(s) = \boldsymbol{c}\left(\boldsymbol{M}s^2 + \boldsymbol{K}\right)^{-1}\boldsymbol{b} = \frac{1}{s^2 + (\eta\omega_0)^2} \ . \tag{3.119}$$

The third state is the only one which contributes to the input-output-behaviour.

The Krylov basis for this system and $m_e = 1$, reads

$$\boldsymbol{V} = \left(\boldsymbol{K} + s_e^2\boldsymbol{M}\right)^{-1}\begin{bmatrix}\boldsymbol{b} & \boldsymbol{c}^T\end{bmatrix} \ , \tag{3.120}$$

whereas $\begin{bmatrix}\boldsymbol{b} & \boldsymbol{c}^T\end{bmatrix}$ builds the starting matrix for the block Krylov algorithm. The reduced system's transfer function can then be calculated analytically by projection with $\boldsymbol{V}$ and the relative error can be calculated according to

$$E_{RK}(s) = \frac{\hat{G}(s) - G(s)}{G(s)} \ . \tag{3.121}$$

The full symbolic expression of the relative error from *Eq.* (3.121) can be shown to be

$$E_{RK}(s) = \frac{\hat{G}(s) - G(s)}{G(s)} = \frac{N_1}{D_1} - 1 \tag{3.122}$$

with the numerator and denominator from *Eq.* (3.123) and *Eq.* (3.124), respectively.

$$\begin{aligned} N_1 = &\left({s_e}^2 + {\omega_0}^2\right)\left(\eta^2\,{\omega_0}^2 + s^2\right)\left(2\,\eta^2\,s^2\,{\omega_0}^2 - \eta^2\,{s_e}^2\,{\omega_0}^2\right. \\ &\left. + \eta^2\,{\omega_0}^4 + 3\,s^2\,{s_e}^2 + s^2\,{\omega_0}^2 + 4\,{s_e}^2\,{\omega_0}^2 + 2\,{\omega_0}^4\right) \end{aligned} \tag{3.123}$$

$$\begin{aligned} D_1 = &\left(s^2 + {\omega_0}^2\right)\left(\eta^4\,s^2\,{\omega_0}^4 + \eta^4\,{\omega_0}^6 + 2\,\eta^2\,s^2\,{s_e}^2\,{\omega_0}^2 + 2\,\eta^2\,{s_e}^4\,{\omega_0}^2\right. \\ &\left. + 6\,\eta^2\,{s_e}^2\,{\omega_0}^4 + 2\,\eta^2\,{\omega_0}^6 + 3\,s^2\,{s_e}^4 + 4\,s^2\,{s_e}^2\,{\omega_0}^2 + 2\,s^2\,{\omega_0}^4 + {s_e}^4\,{\omega_0}^2\right) \end{aligned} \tag{3.124}$$

The most interesting case is when the eigenfrequencies ω_0 and $\eta\omega_0$ are separated as much as possible. This can be analysed with the limit $\eta \longrightarrow \infty$. One can show that the relative error is then

$$\lim_{\eta\to\infty} E_{RK}(s) = \lim_{\eta\to\infty} \frac{\hat{G}(s) - G(s)}{G(s)} = -\frac{\left(s^2 - s_e^2\right)^2}{\left(s^2 + \omega_0^2\right)^2} \ . \tag{3.125}$$

The hypothesis, which is not proven analytically, is that the relative error term multiplies repeatedly for each further Krylov iteration and expansion point as

$$|E_{RK}(s)| \le \prod_{k=1}^{n_e} \left| \frac{\left(s^2 - \boldsymbol{s}_{\boldsymbol{e}_k}^2\right)^{2\boldsymbol{m}_{\boldsymbol{e}_k}}}{\left(s^2 + \omega_0^2\right)^{2\boldsymbol{m}_{\boldsymbol{e}_k}}} \right| . \tag{3.126}$$

The relative FRF error is then obtained by setting $s = i\omega$ as

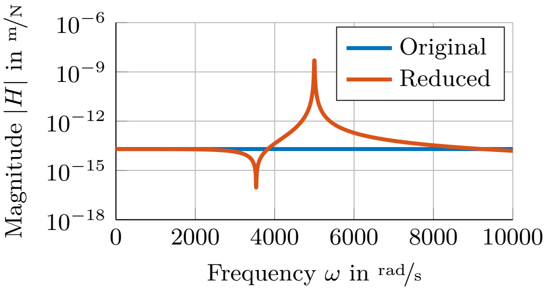
$$|E_{RK}(i\omega)| \le \prod_{k=1}^{n_e} \left| \frac{\left(\omega^2 + \boldsymbol{s}_{\boldsymbol{e}_k}^2\right)^{2\boldsymbol{m}_{\boldsymbol{e}_k}}}{\left(\omega^2 - \omega_0^2\right)^{2\boldsymbol{m}_{\boldsymbol{e}_k}}} \right| = \hat{E}_{RK}(i\omega) . \tag{3.127}$$

3.4.2.2 Numerical Examples

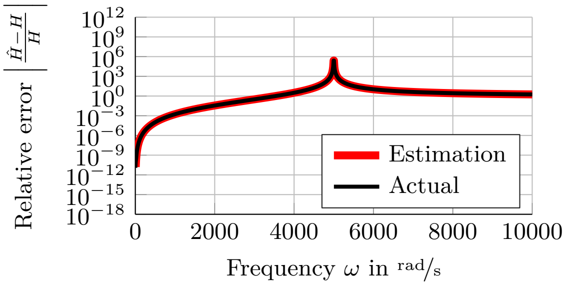
As a numerical example, the system from *Eq.* (3.118) is used with $\omega_0 = 5\,000\,\mathrm{rad}/\mathrm{s}$ and $\eta = 1\,000$ according to

$$\begin{aligned} \boldsymbol{M} &= \boldsymbol{I}\,, \quad \boldsymbol{K} = \operatorname{diag}\left(5\,000^2, 5\,000^2, 5\,000\,000^2\right) , \\ \boldsymbol{b} &= \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^T , \quad \boldsymbol{c} = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} . \end{aligned} \tag{3.128}$$

The FRFs of the original system and the reduced system are shown in *Fig.* 3.11(*a*) for $m_e = 1$, and $s_e = 0$. The relative error for the reduced system and the estimation from *Eq.* (3.127) are shown in *Fig.* 3.11(*b*) and are nearly identical.

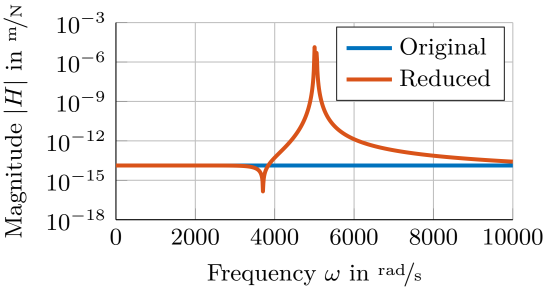


(a) FRFs

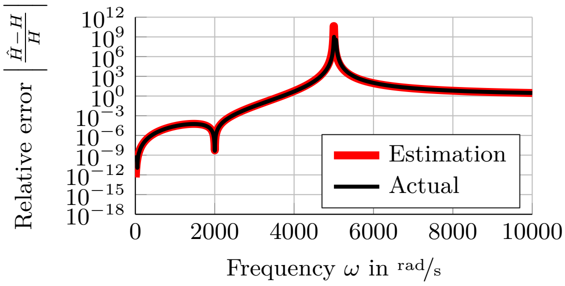


(b) Relative error

Figure 3.11: FRFs and relative error for the system from *Eq.* (3.128) and Krylov reduction with one expansion point $\left(s_e = 0, m_e = 1, n = 2\right)$



(a) FRFs



(b) Relative error

Figure 3.12: FRFs and relative error for the system from *Eq.* (3.129) and Krylov reduction with two expansion points $\left(\boldsymbol{s_e} = [0 \;\; i2\,000], \boldsymbol{m_e} = [1 \;\; 1], n = 4\right)$

Because the system from *Eq.* (3.118) is of third order, matching more than one moment for both the input and the output vector leads to an exact representation of the system. Therefore, a slightly extended system with two more states is used to show the behaviour of the relative error when reducing with multiple expansion points. The system is defined by

$$\begin{aligned}
\boldsymbol{M} &= \boldsymbol{I}\,, \quad \boldsymbol{K} = \operatorname{diag}\left(5\,000^2, 5\,000^2, 5\,050^2, 5\,050^2, 5\,000\,000^2\right)\,, \\
\boldsymbol{b} &= \begin{bmatrix}1 & 0 & 1 & 0 & 1\end{bmatrix}^T\,, \quad \boldsymbol{c} = \begin{bmatrix}0 & 1 & 0 & 1 & 1\end{bmatrix}\,.
\end{aligned} \tag{3.129}$$

This system has four states which are either not observable or not controllable with the frequencies 5 000 rad/s and 5 050 rad/s. The fifth mode has a much higher frequency and is the only state influencing the transfer function.

The resulting relative error for Krylov subspace reduction with one expansion point at 0 rad/s and one at 2 000 rad/s is shown in *Fig.* 3.12 together with the error estimation from *Eq.* (3.127). Again, the error is estimated nearly perfectly.

3.4.2.3 Influence of Zeros Below the First Pole

The FRFs of the systems above do not have any zeros below the first pole but, in general, zeros can occur in this frequency range.

It is known from *Section* 2.5.5 that Krylov reduction leads to a Padé approximation of the transfer function, i.e. a rational function according to

$$\hat{G}(\sigma) = \frac{\alpha_{n-1}(\sigma)}{\beta_n(\sigma)} = \frac{a_0 + a_1\,\sigma + a_2\,\sigma^2 + \ldots + a_{n-1}\,\sigma^{n-1}}{1 + b_1\,\sigma + b_2\,\sigma^2 + \ldots + b_n\,\sigma^n}\,, \tag{3.130}$$

if the subspace dimension is n. Because the system with second-order structure is developed as system with first-order structure according to *Eq.* (3.7), the Laplace parameter is substituted by $\sigma = s^2$. The degree of the numerator polynomial α is $n - 1$ and the degree of the denominator polynomial β is n.

Considering the Krlyov basis from *Eq.* (3.120) with $m_e = 1$ which leads to a subspace dimension of $n = 2$, the transfer function of the reduced system is then of the form

$$\hat{G}(s) = \frac{a_0 + a_1\,\sigma}{1 + b_1\,\sigma + b_2\,\sigma^2} \tag{3.131}$$

and has a zero at

$$\sigma_z = -\frac{a_0}{a_1}\,, \tag{3.132}$$

and thus, a complex conjugate pair of zeros at

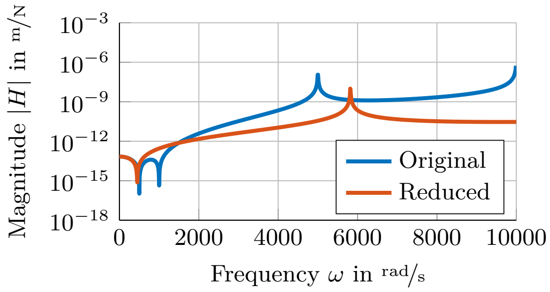
$$s_z = \sqrt{\sigma_z} = \pm i\sqrt{\frac{a_0}{a_1}}\,. \tag{3.133}$$

Looking at the positive frequencies of the FRF, there is exactly one transmission zero.

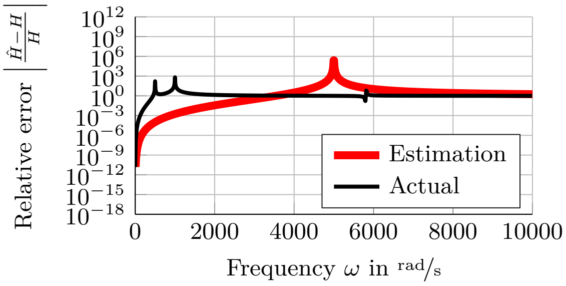
If the original system has more than one zero, all but one of the zeros are completely disregarded. Consider the system

$$\begin{aligned}
\boldsymbol{M} &= \boldsymbol{I}\,, \quad \boldsymbol{K} = \operatorname{diag}\left(5\,000^2, 10\,000^2, 15\,000^2, 20\,000^2, 25\,000^2\right)\,, \\
\boldsymbol{b} &= \begin{bmatrix}0.013 & -0.242 & -0.028 & 0.209 & 0.947\end{bmatrix}^T\,, \\
\boldsymbol{c} &= \begin{bmatrix}0.447 & 0.447 & 0.447 & 0.447 & 0.447\end{bmatrix}\,,
\end{aligned} \tag{3.134}$$

with the first pole at 5 000 rad/s and two zeros, one at 500 rad/s and one at 1 000 rad/s. The FRFs for the original and the reduced system as well as the relative error are shown in *Fig.* 3.13. One can see that



(a) FRFs



(b) Relative error

Figure 3.13: FRFs and relative error for the system from *Eq.* (3.134), exhibiting two zeros below the first pole $\left(s_e = 0, m_e = 1, n = 2\right)$

only one zero is modelled with the reduced system and that, therefore, the approximation is useless from the frequency of the first zero on. The actual relative error clearly exceeds the estimation.

To overcome this issue, two possibilities are proposed. One is to enhance the number of Krylov iterations m_e in order to increase the degree of the numerator polynomial. This is shown in *Fig.* 3.14 for the same system but with $m_e = 2$. The relative error is then below the estimated error bound, with the exception of the zero frequencies, which lead to poles in the relative error. Note that the error bound is the error bound calculated with $m_e = 2$ and is thus lower than the error bound in *Fig.* 3.13.

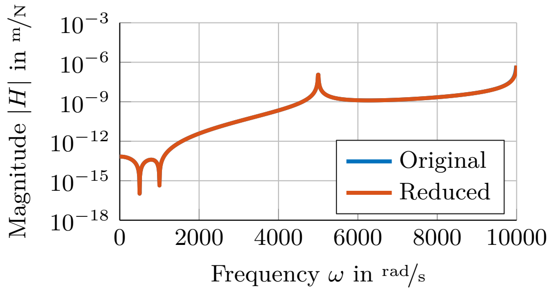
Another possibility is to explicitly add the zero shape vectors $\boldsymbol{\phi}_{\mathbf{z}_k}$ to the projection basis. The zero shapes can be calculated according to *Eq.* (3.32) for a scalar transfer function, i.e. for one input to one output. Extending the basis used for *Fig.* 3.13 with the zero shape vectors leads to the FRFs and the relative error shown in *Fig.* 3.15. The resulting relative error lies clearly below the estimation. Because the solution of an eigenvalue problem for a system of the dimension $N + 1$, i.e. the original system's dimension enhanced by one, is required for each input and output combination, this method is only useful for SISO systems or systems with a very small number of inputs and outputs.

3.4.2.4 Verification of the Estimation of the Relative Error Using Random Systems

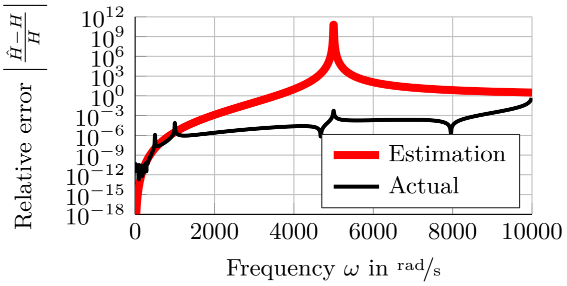
To demonstrate the usefulness of the error estimator from *Eq.* (3.127), the estimation is tested with random systems generated according to *Section* 3.1.5.1. The FRFs of 20 randomly produced systems are shown in *Fig.* 3.16.

All modes with eigenfrequencies below $\omega_m = 5\,000\,\mathrm{rad/s}$ are used for modal reduction, and therefore, the remainder system after modal reduction has no poles below this frequency. The FRFs of the remainder systems are shown in *Fig.* 3.17.

These remainder systems are then reduced with Krylov reduction with different settings. The results of the case with one expansion point are shown in *Fig.* 3.18. This figure shows a good quality of the estimation of the relative error. However, the actual relative error exceeds the estimation for some systems. For the yellow line, e.g., this is due to a zero at $s = \pm 79\,\mathrm{rad/s}$. This zero is real-valued and

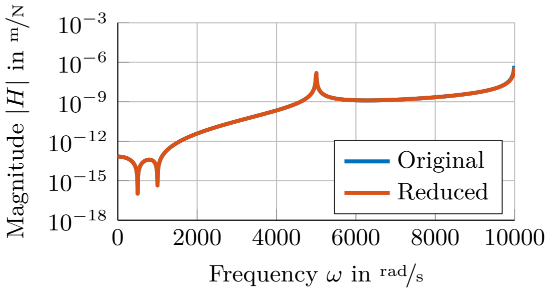


(a) FRFs

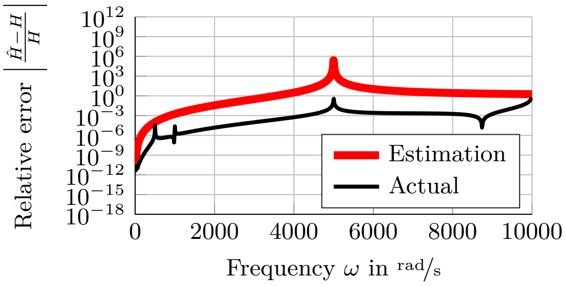


(b) Relative error

Figure 3.14: FRFs and relative error for the same example system as in *Fig.* 3.13 but with an extended Krylov basis $(s_e = 0, m_e = 2, n = 4)$



(a) FRFs



(b) Relative error

Figure 3.15: FRFs and relative error for the same example system as in *Fig.* 3.13 but with additional zero shapes in the projection basis $(s_e = 0, m_e = 2, n_z = 1, n = 4)$

thus not a zero of the FRF but, nevertheless, a zero below the first pole and thus leads to problems as described in *Section* 3.4.2.3.

The case with two expansion points and one Krylov iteration each is shown in *Fig.* 3.19. This case shows a very good estimation of the upper bound for the relative error.

The third case from *Fig.* 3.20 shows the approximation of the random systems with two Krylov iterations at an expansion point unequal to zero. Again, the overall estimation of the error is good, small issues arise at very low frequencies due to the described transfer function zero.

Overall, good results can be achieved with the proposed estimation for the relative error bound.

3.4.3 Estimation of the Relative Error for Modal Subspace Based Reduction

The remainder FRFs shown in *Fig.* 3.17 are, by definition, the absolute error FRFs after modal reduction. Building the relative error means dividing the absolute error by the original FRFs. Because the remainder FRFs have no poles in the frequency range below ω_m, the poles of the original transfer function become zeros of the relative error and vice versa. Usually, the mean value of the relative error after modal reduction is smaller than 1 in the frequency range of the considered modes.

As a rough estimation, it is supposed that the relative error is below 1, except in the neighbourhood of the zeros of the original transfer function, i.e.

$$|E_M(i\omega)| \leq 1 = \hat{E}_M(i\omega)\,, \quad \omega \in \Omega_l - \Omega_z\,. \tag{3.135}$$

In *Fig.* 3.21, the relative error and the estimation are shown for the random systems. The modes for the reduction are considered up to 5 000 rad/s and up to this frequency, only separated peaks exceed the estimation.

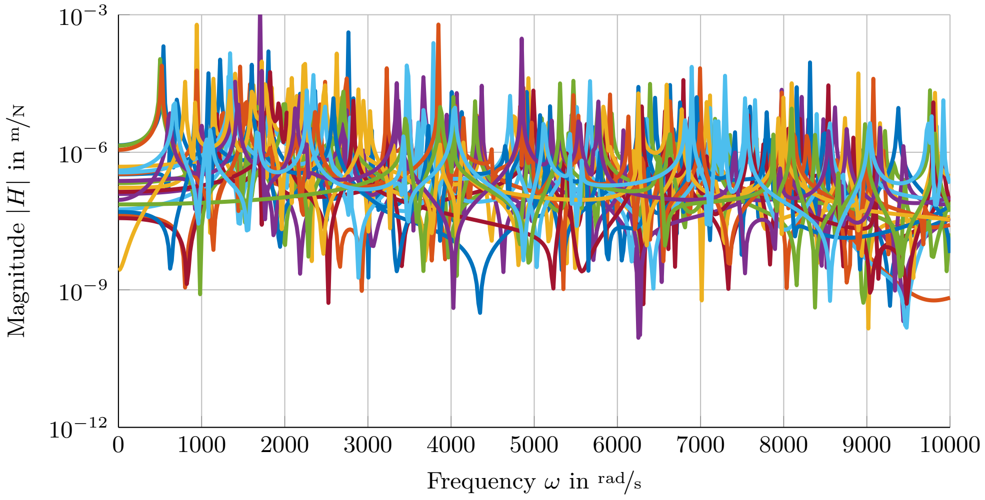


Figure 3.16: FRFs of 20 randomly generated systems according to *Section* 3.1.5.1

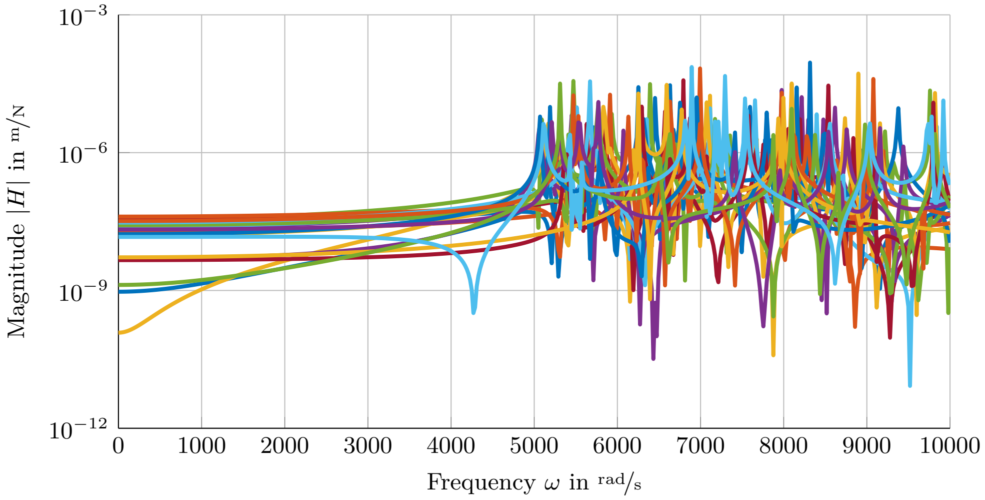


Figure 3.17: Remainder FRFs $H_R(i\omega)$ after modal reduction according to *Section* 3.3.1 of the systems from *Fig.* 3.16, where all eigenfrequencies below $\omega_m = 5\,000$ rad/s were taken into account

3.4.4 Estimation of the Relative Error for KMS Based Reduction

From *Eq.* (3.111) it is known that the relative error for KMS reduced systems can be separated into a product of the relative error of the modally reduced system and the relative error of the Krylov reduced remainder system. Therefore, as an estimation for the upper bound of the total relative error E_r, the product of the upper bound estimations for both factors is used, i.e.

$$|E_r(i\omega)| \leq \hat{E}_M(i\omega)\,\hat{E}_{RK}(i\omega) \leq \prod_{k=1}^{n_s} \left| \frac{\left(\omega^2 + s_{e_k}^2\right)^{2m_k}}{\left(\omega^2 - \omega_m^2\right)^{2m_k}} \right| = \hat{E}_r(i\omega)\ . \tag{3.136}$$

Fig. 3.22 to 3.24 show the relative error for the systems reduced using KMS projection, with the Krylov subspaces used for *Fig.* 3.18 to 3.20 and the modal subspaces used for *Fig.* 3.21. In contrast to the relative error of the Krylov reduced remainder system, the FRFs are not smooth in the frequency range below ω_m anymore. However, the estimated error bound holds very well for all FRFs, only at the poles of the relative error, i.e. the zeros of the original transfer function, the limit is exceeded locally.

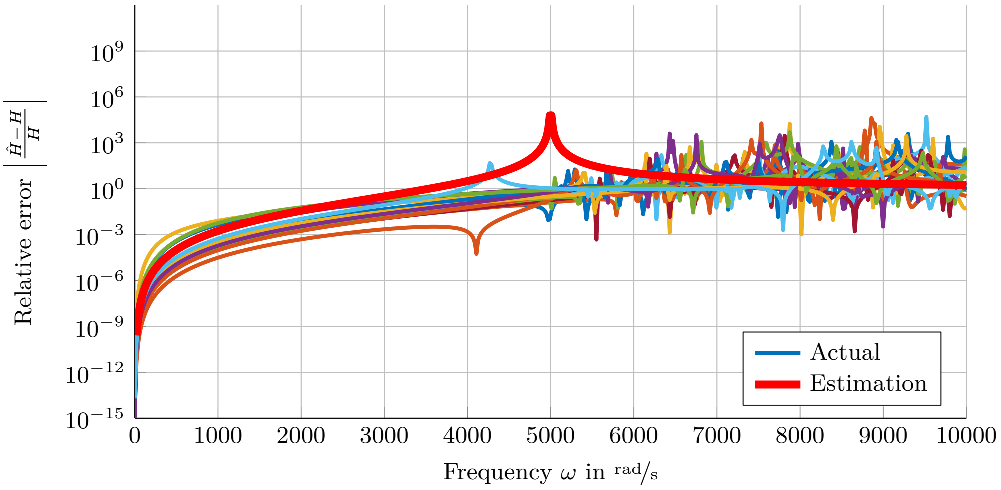


Figure 3.18: Relative error of the FRFs after Krylov based reduction of the remainder systems from *Fig.* 3.17 with one expansion point $\left(s_e = 0, m_e = 1, n = 2\right)$

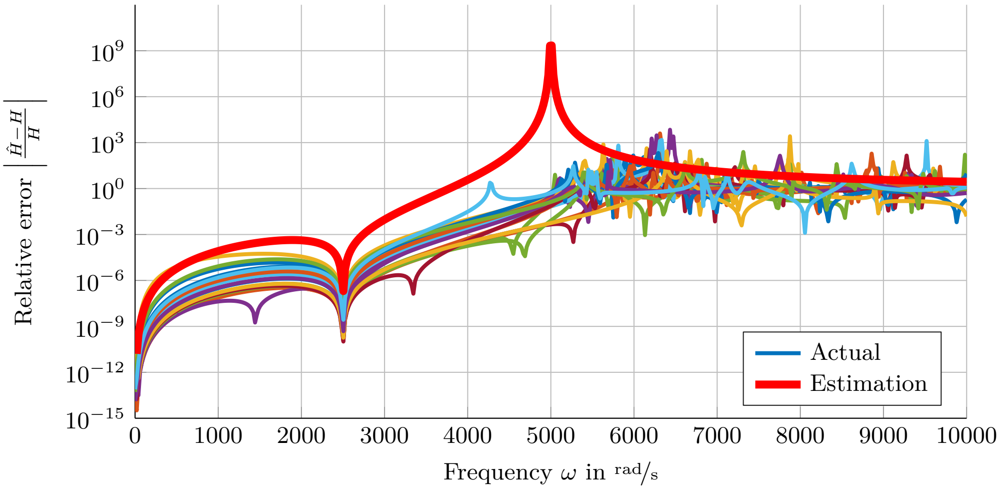


Figure 3.19: Relative error of the FRFs after Krylov based reduction of the remainder systems from *Fig.* 3.17 with two expansion points $\left(\boldsymbol{s_e} = [0 \;\; i\,2\,500], \boldsymbol{m_e} = [1 \;\; 1], n = 4\right)$

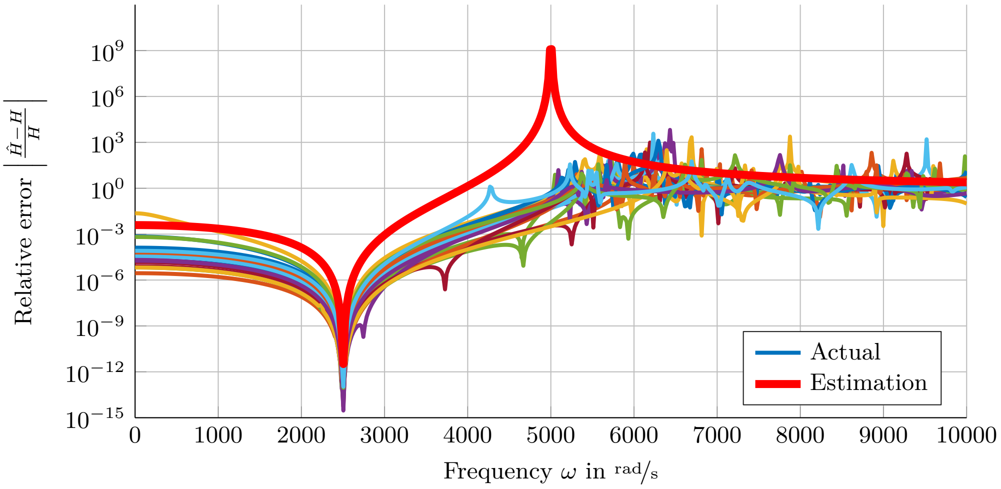


Figure 3.20: Relative error of the FRFs after Krylov based reduction of the remainder systems from *Fig.* 3.17 with one expansion point and two Krylov iterations $\left(s_e = i\,2\,500, m_e = 2, n = 4\right)$

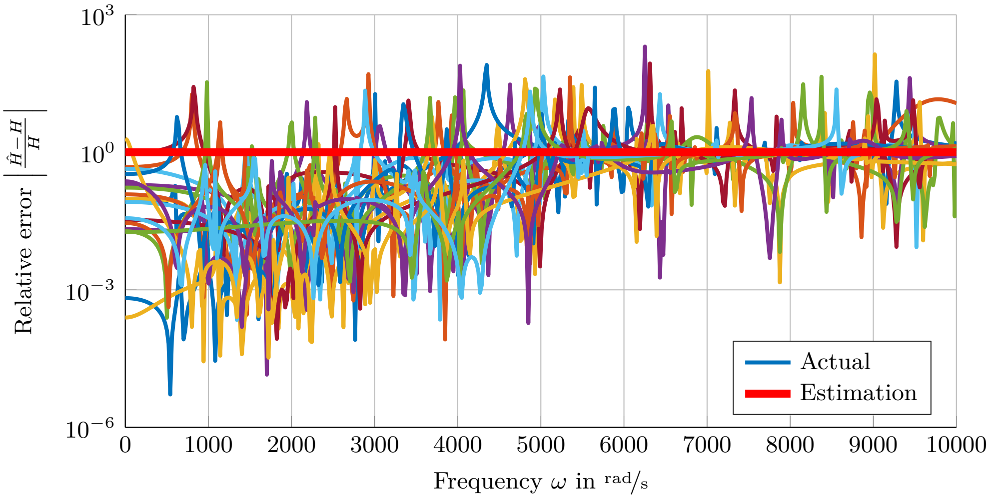


Figure 3.21: Relative error of the FRFs after modal reduction, i.e. for the remainder systems from *Fig.* 3.17, with $\omega_m = 5\,000\,\mathrm{rad/s}$

3.4.5 Estimation of the Shift of Transmission Zeros

In *Eq.* (3.33), it was shown that a zero in the transfer can be seen as cancellation of two partial FRFs H_1 and H_2 according to

$$H(i\omega) = H_1(i\omega) + H_2(i\omega) = \frac{\tilde{\boldsymbol{c}}_1\,\tilde{\boldsymbol{b}}_1}{\omega_1^2 - \omega^2} + \frac{\tilde{\boldsymbol{c}}_2\,\tilde{\boldsymbol{b}}_2}{\omega_2^2 - \omega^2}\ . \tag{3.137}$$

An example for such a system is shown in *Fig.* 3.25 together with the asymptotes for $H_1(i\omega)\big|_{\omega\to\infty}$ and $H_2(i\omega)\big|_{\omega\to 0}$ for the parameters

$$\omega_1 = 10\,,\quad \omega_2 = 2\,000\,,\quad \boldsymbol{b} = \begin{bmatrix}1 & 1\end{bmatrix}^T\,,\quad \boldsymbol{c} = \begin{bmatrix}1 & 1\,000\end{bmatrix}\ . \tag{3.138}$$

The relative error arising from reduction leads to a scaling of the magnitude of the partial systems from *Eq.* (3.137) and the zero frequency equates to the intersection frequency between H_1 and H_2. By scaling both partial systems in an unfavourable way, one can write the transfer function of the perturbed system as

$$\hat{H}(i\omega) = \delta H_1(i\omega) + \frac{1}{\delta}H_2(i\omega) = \delta\frac{\tilde{\boldsymbol{c}}_1\,\tilde{\boldsymbol{b}}_1}{\omega_1^2 - \omega^2} + \frac{1}{\delta}\frac{\tilde{\boldsymbol{c}}_2\,\tilde{\boldsymbol{b}}_2}{\omega_2^2 - \omega^2}\,, \tag{3.139}$$

where $\delta = 1 + \hat{E}_r$ is the scaling factor derived from the relative magnitude error bound.

Scaling a transfer function leads in logarithmic representation to a shift of the curve in magnitude, and thus, to a shift of the intersection point in frequency. The highest sensitivity for the shift of the intersection point in frequency arises when the intersection lies in the area where the magnitudes run out asymptotically. Therefore, the intersection point of the asymptotes according to

$$\delta\frac{\boldsymbol{c}_1\,\boldsymbol{b}_1}{\omega^2} = \frac{1}{\delta}\frac{\boldsymbol{c}_2\,\boldsymbol{b}_2}{\omega_2^2} \tag{3.140}$$

is analysed here. The zero frequency of the perturbed system reads then

$$\hat{\omega}_z = \delta\sqrt{\frac{\tilde{\boldsymbol{c}}_1\,\tilde{\boldsymbol{b}}_1}{\tilde{\boldsymbol{c}}_2\,\tilde{\boldsymbol{b}}_2}}\,\omega_2\,, \tag{3.141}$$

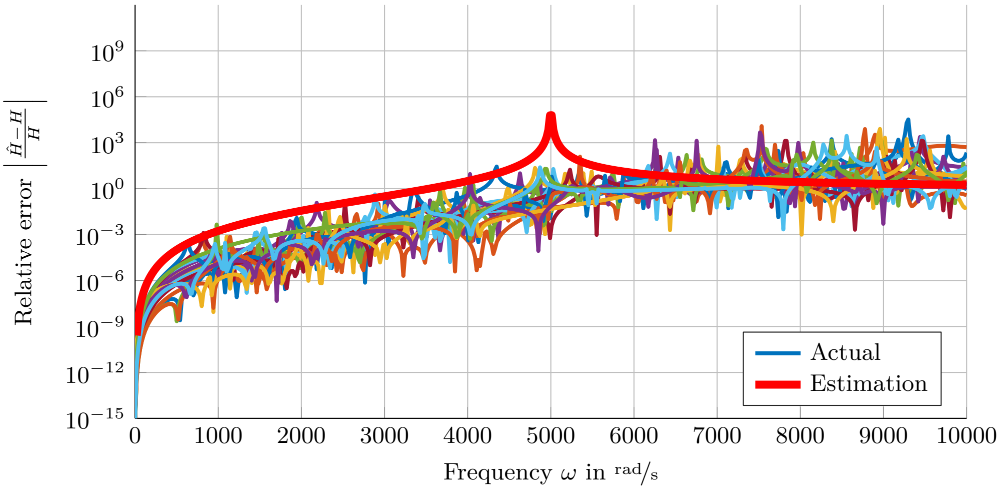


Figure 3.22: Relative error of the FRFs after KMS based reduction of the systems from *Fig.* 3.16 with one expansion point $\left(s_e = 0, m = 1, \omega_m = 5\,000\right)$

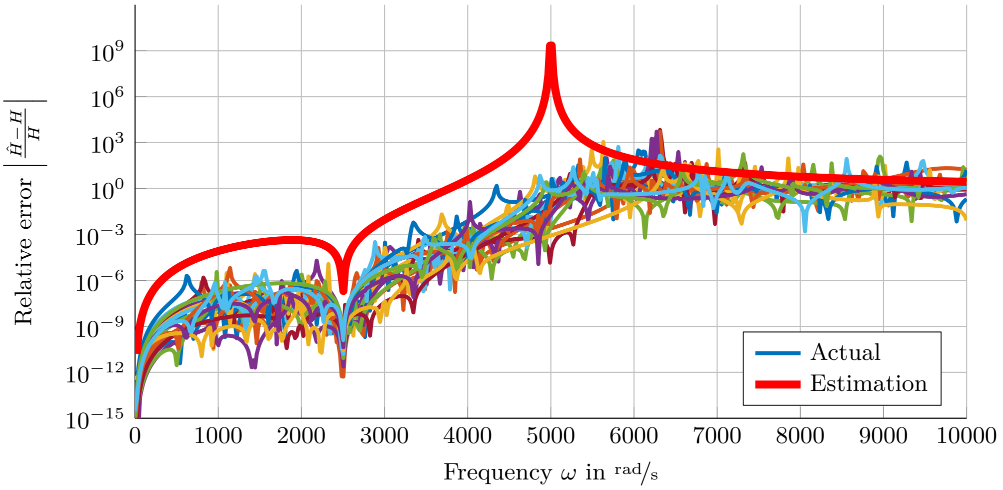


Figure 3.23: Relative error of the FRFs after KMS based reduction of the systems from *Fig.* 3.16 with two expansion points $\left(\boldsymbol{s_e} = [0 \;\; i\,2\,500], \boldsymbol{m} = [1 \;\; 1], \omega_m = 5\,000\right)$

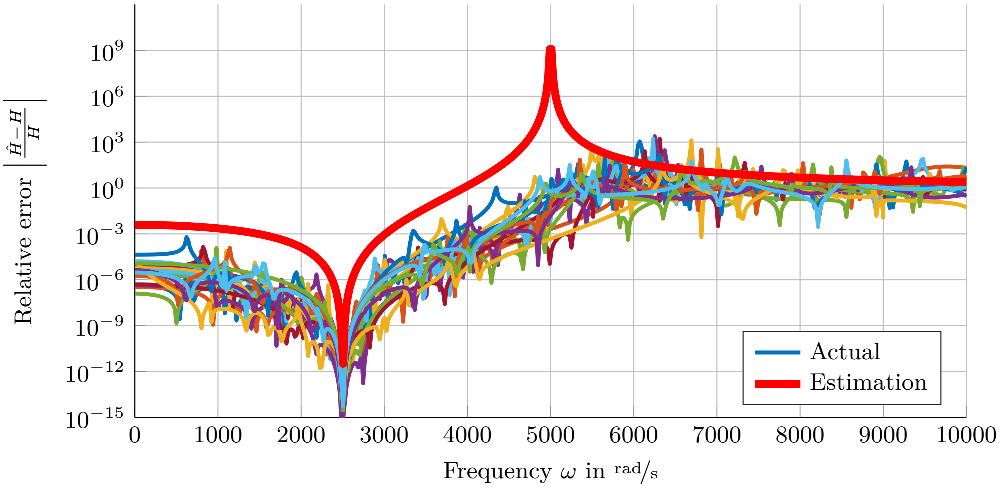


Figure 3.24: Relative error of the FRFs after KMS based reduction of the systems from *Fig.* 3.16 with one expansion point and two Krylov iterations $\left(s_e = i\,2\,500, m = 2, \omega_m = 5\,000\right)$

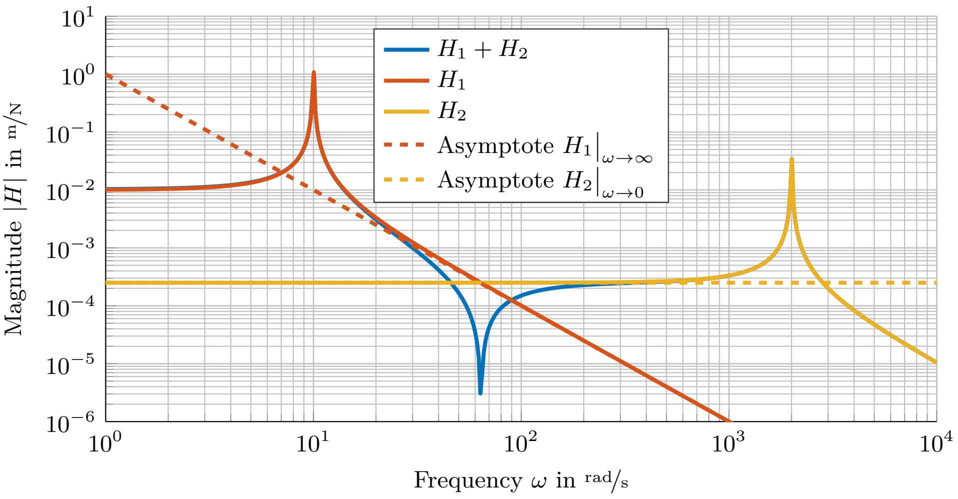


Figure 3.25: Example of an FRF ($H_1 + H_2$) exhibiting a zero due to cancellation of two partial FRFs (H_1 and H_2) and the asymptotes $H_1|_{\omega\to\infty}$ and $H_2|_{\omega\to 0}$

and, by setting $\delta = 1$, the zero frequency for the original system reads

$$\omega_z = \sqrt{\frac{\tilde{\boldsymbol{c}}_1\,\tilde{\boldsymbol{b}}_1}{\tilde{\boldsymbol{c}}_2\,\tilde{\boldsymbol{b}}_2}}\,\omega_2\ . \tag{3.142}$$

This shows that the maximum relative zero shift is proportional to the maximum magnitude scaling according to

$$\hat{\omega}_z = \delta\,\omega_z\ . \tag{3.143}$$

The relative zero error can then be calculated to

$$\frac{\hat{\omega}_z - \omega_z}{\omega_z} = \frac{\delta\,\omega_z - \omega_z}{\omega_z} = \delta - 1 = \hat{E}_r\ , \tag{3.144}$$

what exactly matches the relative magnitude error.

The relative zero error is investigated for the KMS based reduced system with the relative error as shown in *Fig.* 3.22 and is illustrated in *Fig.* 3.26. This shows that the estimation for the relative zero error is appropriate as an estimation for the upper bound.

3.4.6 Conclusions on the Error Estimation

In *Section* 3.2.2, the requirements on the accuracy have been specified for model order reduction. The upper bounds

- $\bar{\varepsilon}_r$, for the relative error of the FRFs,
- $\bar{\varepsilon}_z$, for the relative error of the transmission zeros, and
- $\bar{\varepsilon}_p$, for the relative error of the poles

have been defined for the frequency range of interest. Furthermore, the static behaviour is required to match accurately.

For the relative error of the FRF and the relative error of the transmission zeros, an estimation for an upper bound has been presented in *Eq.* (3.136), which reads

$$\hat{E}_r(i\omega) = \prod_{k=1}^{n_s} \left| \frac{\left(\omega^2 + \boldsymbol{s}_{\boldsymbol{e}_k}^2\right)^{2\boldsymbol{m}_{\boldsymbol{e}_k}}}{\left(\omega^2 - \omega_m^2\right)^{2\boldsymbol{m}_{\boldsymbol{e}_k}}} \right|\ , \tag{3.145}$$

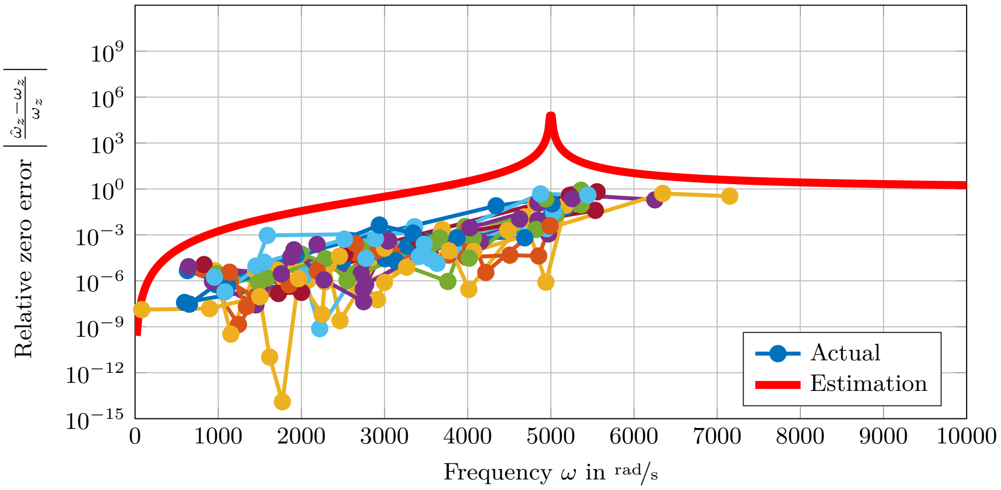


Figure 3.26: Relative error of all zero frequencies of the KMS based reduced systems from *Fig.* 3.22 $(s_e = 0, m_e = 1, \omega_m = 5\,000)$

where $\boldsymbol{s_e} \in \mathbb{C}^{n_e}$ is a vector with expansion points for the Krylov subspace, $\boldsymbol{m_e} \in \mathbb{Z}^{n_e}$ a vector with the number of Krylov iterations per expansion point, and ω_m the limit frequency up to which the modes are taken into account for modal reduction.

Due to the inclusion of eigenvectors in the projection basis of a KMS, the corresponding poles of the reduced system match exactly.

For defined limits $\bar{\varepsilon}_r$, $\bar{\varepsilon}_z$, and $\bar{\varepsilon}_p$, a feasible set of reduction parameters can be found a priori of the reduction by considering following criteria. At least all eigenvectors with eigenfrequencies within the frequency range of interest, i.e. $\omega_k \in \Omega_l$, have to be added to the reduction basis in order to match the poles, and therefore

$$\omega_m \geq \omega_l \ . \tag{3.146}$$

The estimated relative error must lie below the limit for the relative error of the FRF and the one of the transmission zeros, according to

$$\hat{E}_r(i\omega) \leq \min\{\bar{\varepsilon}_r, \bar{\varepsilon}_z\} \ , \quad \forall \omega \in \Omega_l \ . \tag{3.147}$$

In order to match the static behaviour, one expansion point has to be chosen at zero, i.e. $0 \in \boldsymbol{s_e}$.

For systems with a moderate to large number of inputs and outputs, it is beneficial to keep the number of Krylov iterations low in order to achieve reduced systems of low order, e.g. set $s_e = 0$ and $m_e = 1$ with ω_m fulfilling *Eq.* (3.146) and *Eq.* (3.147).

3.5 Implementation

3.5.1 A KMS Algorithm

A basis for a KMS according to *Section* 3.3.3 contains, by definition, eigenvectors and a Krylov basis for a SISO or MIMO system and possibly for multiple expansion points. An algorithm for the creation of a KMS basis is presented in *Algorithm* 1. In order to consider all inputs and outputs, a combined input and output matrix $\boldsymbol{B_c}$ is created in *Line* 3. Through the EXTENDRANGE procedure described in *Algorithm* 6, an orthonormal matrix with

$$\mathrm{range}(\boldsymbol{B_c}) = \mathrm{range}(\boldsymbol{B}) + \mathrm{range}(\boldsymbol{C}^T) \tag{3.148}$$

is created, whereas input and output vectors that are linear dependent will be automatically dropped. In *Line* 7, a basis for a modal subspace according to *Section* 3.3.1, and in *Line* 8, a basis for a Krylov subspace according to *Section* 3.3.2 is created for the undamped system. In *Line* 9, both bases are combined to one orthonormal basis.

Algorithm 1 KMS basis creation

```
 1: procedure KMS($\boldsymbol{M}, \boldsymbol{K}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{s_e}, \boldsymbol{m_e}, \omega_m$)
 2:     $\boldsymbol{M}, \boldsymbol{K} \in \mathbb{R}^{N \times N}$, $\boldsymbol{B} \in \mathbb{R}^{N \times p}$, $\boldsymbol{C} \in \mathbb{R}^{q \times N}$    ▷ System matrices in second-order structure
 3:     $\boldsymbol{B_c}$ := ExtendRange($\boldsymbol{B}, \boldsymbol{C}^T$)    ▷ Combine all inputs and outputs for the Krylov iteration
 4:     $\boldsymbol{s_e} \in \mathbb{C}^{n_e}$    ▷ Expansion points
 5:     $\boldsymbol{m_e} \in \mathbb{Z}^{n_e}$    ▷ Number of Krylov iterations per expansion point
 6:     $\omega_m \in \mathbb{R}_+$    ▷ Limit frequency for eigenvalues to keep
 7:     $\boldsymbol{\Phi_M}$ := Eigs($-\boldsymbol{K}, \boldsymbol{M}, \omega_m^2$)    ▷ Solve eigenvalue problem
 8:     $\boldsymbol{V_K}$ := RationalArnoldi($-\boldsymbol{K}, \boldsymbol{M}, \boldsymbol{B_c}, \boldsymbol{m_e}, \boldsymbol{s_e}^2$)    ▷ Create Krylov basis
 9:     $\boldsymbol{V}$ := ExtendRange($\boldsymbol{V_K}, \boldsymbol{\Phi_M}$)    ▷ Combine the modal and Krylov basis
10:     return $\boldsymbol{V}$    ▷ Orthonormal KMS basis
11: end procedure
```

As specified in *Section* 3.2.3, due to its numerical robustness, the Arnoldi algorithm is used for the creation of Krylov subspace bases. The RationalArnoldi procedure is presented in *Algorithm* 2 for the creation of a Krylov subspace basis for multiple expansion points. The particular Krylov bases for single expansion points are calculated in *Line* 6 and are then combined in *Line* 10 to a complete orthonormal basis.

Algorithm 2 Krylov subspace basis creation for multiple expansion points

```
 1: procedure RationalArnoldi($\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{\sigma_e}, \boldsymbol{m_e}$)
 2:     $\boldsymbol{A}, \boldsymbol{E} \in \mathbb{R}^{N \times N}$, $\boldsymbol{B} \in \mathbb{R}^{N \times p}$    ▷ System matrices in frist-order structure
 3:     $\boldsymbol{\sigma_e} \in \mathbb{C}^{n_e}$    ▷ Expansion points
 4:     $\boldsymbol{m_e} \in \mathbb{Z}^{n_e}$    ▷ Number of Krylov iterations per expansion point
 5:     for $i := 1, 2, \ldots, n_e$ do    ▷ Repeat for all expansion points
 6:         $\boldsymbol{V}_i$ := BlockArnoldi($\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{m}_{\boldsymbol{e}_i}, \boldsymbol{\sigma}_{\boldsymbol{e}_i}$)    ▷ Krylov basis for expansion point $\boldsymbol{\sigma}_{\boldsymbol{e}_i}$
 7:         if $i = 1$ then
 8:             $\boldsymbol{V} := \boldsymbol{V_i}$    ▷ Initialise basis $\boldsymbol{V}$
 9:         else
10:             $\boldsymbol{V}$ := ExtendRange($\boldsymbol{V}, \boldsymbol{V_i}$)    ▷ Extend basis $\boldsymbol{V}$ by $\boldsymbol{V}_i$
11:         end if
12:     end for
13:     return $\boldsymbol{V}$    ▷ Orthonormal rational Krylov subspace basis
14: end procedure
```

The BlockArnoldi procedure which is repeatedly called in *Algorithm* 2 is described in *Algorithm* 3. For numerical stability, the orthonormalisation of the basis in each iteration step in *Line* 8 and *Line* 12 is crucial. The ReduceRange procedure creates an orthonormal basis with

$$\operatorname{range}(\boldsymbol{V}_i) = \operatorname{range}(\hat{\boldsymbol{V}}_i) - \operatorname{range}(\boldsymbol{V}) \tag{3.149}$$

In the Orth procedure presented in *Algorithm* 5, a modified Gram-Schmidt algorithm with deflation detection is implemented. Deflation occurs when one column is not linearly independent of the existing columns of the matrix $\boldsymbol{Q}$ in *Line* 11. For the detection of deflation, a numerical tolerance ϵ is defined.

Algorithm 3 Krylov subspace basis creation for one expansion point

```
 1: procedure BlockArnoldi($\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \sigma_e, m_e$)
 2:     $\boldsymbol{A}, \boldsymbol{E} \in \mathbb{R}^{N\times N}, \boldsymbol{B} \in \mathbb{R}^{N\times p}$        ▷ System matrices in frist-order structure
 3:     $\sigma_e \in \mathbb{C}$                                               ▷ Expansion point
 4:     $m_e \in \mathbb{Z}$                                               ▷ Number of Krylov iterations
 5:     for $i := 1, 2, \ldots, m_e$ do                        ▷ Repeat for the desired number of iterations
 6:         if $i = 1$ then
 7:             $\hat{\boldsymbol{V}}_1 := (\boldsymbol{A} - \sigma_e \boldsymbol{E})^{-1} \boldsymbol{B}$                               ▷ First iteration
 8:             $\boldsymbol{V}_1 :=$ Orth($\hat{\boldsymbol{V}}_1$)                            ▷ Orthonormalise the basis
 9:             $\boldsymbol{V} := \boldsymbol{V}_1$                                       ▷ Initialise basis $\boldsymbol{V}$
10:         else
11:             $\hat{\boldsymbol{V}}_i := (\boldsymbol{A} - \sigma_e \boldsymbol{E})^{-1} \boldsymbol{E}\, \boldsymbol{V}_{i-1}$                      ▷ Krylov iteration
12:             $\boldsymbol{V}_i :=$ ReduceRange($\hat{\boldsymbol{V}}_i, \boldsymbol{V}$)             ▷ Orthogonalise to existing basis
13:             $\boldsymbol{V} :=$ ExtendRange($\boldsymbol{V}, \boldsymbol{V}_i$)                      ▷ Extend basis $\boldsymbol{V}$ by $\boldsymbol{V}_i$
14:         end if
15:     end for
16:     return $\boldsymbol{V}$                                  ▷ Orthogonal Krylov subspace basis
17: end procedure
```

3.5.2 Numerical Implementation

The large-scale matrix in *Line* 7 and 11 of *Algorithm* 3 can, in general, not be inverted directly. However, it is sufficient to solve the systems of equations

$$(\boldsymbol{A} - \sigma_e \boldsymbol{E})\, \hat{\boldsymbol{V}}_1 = \boldsymbol{B}\,, \text{ and} \tag{3.150}$$

$$(\boldsymbol{A} - \sigma_e \boldsymbol{E})\, \hat{\boldsymbol{V}}_i = \boldsymbol{E}\, \boldsymbol{V}_{i-1} \tag{3.151}$$

for the given right-hand sides $\boldsymbol{B}$ and $\boldsymbol{E}\boldsymbol{V}_{i-1}$. The solution of large-scale sparse systems of equations is much more efficient than the inversion of the matrix. For the repeated solution with the same left-hand side, the most efficient way is to factorise the matrix as LU-decomposition or, for Hermitian, positive definite matrices, as Cholesky decomposition (LL^T) and then use the factors repeatedly for the solution with different right-hand sides by forward and backward substitution. Efficient implementations of the factorisation of large sparse matrices exist. In this thesis, SuperLU [66] is used via SciPy [52].

For the solution of the eigenvalue problem formulated in *Algorithm* 4, it is recommended to use specialised program libraries like, e.g. ARPACK [65], which is used, e.g. by SciPy. ARPACK uses an Arnoldi algorithm to find eigenvalues and eigenvectors near an expansion point, and thus, needs to factorise a matrix of the form $(\boldsymbol{A} - \sigma_e \boldsymbol{E})$ for the Arnoldi iteration. SciPy allows to explicitly hand over an inversion operator to the eigenvalue solver which then calls ARPACK. Therefore, the same matrix factorisation can be used for both the Arnoldi iteration with an expansion point near zero and for the eigenvalue solution. This makes the algorithm very efficient. In the case that only one expansion point near zero is used, solely one factorisation of a large-scale matrix has to be performed.

3.5.3 Selection of Expansion Points

The matrix $(\boldsymbol{A} - \sigma_e \boldsymbol{E})$ from *Algorithm* 3 is singular if an expansion point σ_e is an eigenvalue of the system and can then not be factorised.

In order to match the static behaviour, one expansion point has to be chosen at zero, i.e. $0 \in \boldsymbol{s_e}$. However, if the system is not bonded, the stiffness matrix is singular with six rigid body modes (eigenvalues at zero) and, therefore, also the matrix $\left(-\boldsymbol{K} - s_e^2 \boldsymbol{M}\right) \mathrel{\widehat{=}} (\boldsymbol{A} - \sigma_e \boldsymbol{E})$. Because the mass matrix is positive definite, this problem can be avoided by choosing an expansion point different from (but near to) zero. Since the eigenvalues of the undamped system are imaginary, one can choose a real

Algorithm 4 Eigenvalue problem

```
procedure EIGS($\boldsymbol{A}, \boldsymbol{E}, \sigma_m$)
    $\boldsymbol{A}, \boldsymbol{E} \in \mathbb{R}^{N \times N}$,    ▷ System matrices in frist-order structure
    $\sigma_m \in \mathbb{R}_+$    ▷ Limit value for eigenvalues to keep
    $\boldsymbol{A}\boldsymbol{\phi}_i = \boldsymbol{E}\boldsymbol{\phi}_i\sigma_i$    ▷ Find the eigenvectors with the smallest $n_M$ eigenvalues $|\sigma_i| < \sigma_m$
    $\boldsymbol{\Phi_M} = \begin{bmatrix}\boldsymbol{\phi}_0 & \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{n_M}\end{bmatrix}$    ▷ Eigenvectors
    return $\boldsymbol{\Phi_M}$    ▷ Modal matrix
end procedure
```

Algorithm 5 Matrix orthonormalisation

```
procedure ORTH($\boldsymbol{V}$)
    $\boldsymbol{V} \in \mathbb{R}^{N \times n}$    ▷ Matrix which has to be orthonormalised
    $\epsilon \in \mathbb{R}_+$    ▷ Numerical tolerance, e.g. $\epsilon = 10^{-8}$
    for $i := 1, 2, \ldots, n$ do    ▷ Repeat for all columns of $\boldsymbol{V}$
        $\boldsymbol{v} := \boldsymbol{V}_{:i}/|\boldsymbol{V}_{:i}|$    ▷ Column of $\boldsymbol{V}$. Normalisation is crucial in order to detect deflation
        if $i = 1$ then
            $\boldsymbol{Q} := \boldsymbol{v}$    ▷ Initialise matrix $\boldsymbol{Q}$
        else
            $\boldsymbol{t} := \boldsymbol{Q}^T\boldsymbol{v}$    ▷ Determine collinear parts of columns of $\boldsymbol{Q}$ to $\boldsymbol{v}$
            $\boldsymbol{v} := \boldsymbol{v} - \boldsymbol{Q}\boldsymbol{t}$    ▷ Eliminate parts of $\boldsymbol{v}$ which are collinear to $\boldsymbol{Q}$
            if $|\boldsymbol{v}| > \epsilon$ then    ▷ Check for deflation
                $\boldsymbol{Q} := \begin{bmatrix}\boldsymbol{Q} & \boldsymbol{v}/|\boldsymbol{v}|\end{bmatrix}$    ▷ If not deflated, normalise $\boldsymbol{v}$ and add to $\boldsymbol{Q}$
            end if
        end if
    end for
    return $\boldsymbol{Q}$    ▷ Orthonormal matrix with range($\boldsymbol{Q}$) = range($\boldsymbol{V}$)
end procedure
```

Algorithm 6 Combination of basis matrices to an orthonormal basis with extended range

```
procedure EXTENDRANGE($\boldsymbol{V}, \boldsymbol{W}$)
    $\boldsymbol{V} \in \mathbb{R}^{N \times n_v}, \boldsymbol{W} \in \mathbb{R}^{N \times n_w}$    ▷ Matrices which have to be combined
    $\epsilon \in \mathbb{R}_+$    ▷ Numerical tolerance, e.g. $\epsilon = 10^{-8}$
    $\boldsymbol{Q} :=$ ORTH($\boldsymbol{V}$)    ▷ Orthonormalise the input matrix
    for $i := 1, 2, \ldots, n_w$ do    ▷ Repeat for all columns of $\boldsymbol{W}$
        $\boldsymbol{w} := \boldsymbol{W}_{:i}/|\boldsymbol{W}_{:i}|$    ▷ Column of $\boldsymbol{W}$. Normalisation is crucial in order to detect deflation
        $\boldsymbol{t} := \boldsymbol{Q}^T\boldsymbol{w}$    ▷ Determine collinear parts of columns of $\boldsymbol{Q}$ to $\boldsymbol{w}$
        $\boldsymbol{w} := \boldsymbol{w} - \boldsymbol{Q}\boldsymbol{t}$    ▷ Eliminate parts of $\boldsymbol{w}$ which are collinear to $\boldsymbol{Q}$
        if $|\boldsymbol{w}| > \epsilon$ then    ▷ Check for deflation
            $\boldsymbol{Q} := \begin{bmatrix}\boldsymbol{Q} & \boldsymbol{w}/|\boldsymbol{w}|\end{bmatrix}$    ▷ If not deflated, normalise $\boldsymbol{w}$ and add to $\boldsymbol{Q}$
        end if
    end for
    return $\boldsymbol{Q}$    ▷ Orthonormal matrix with range($\boldsymbol{Q}$) = range($\boldsymbol{V}$) + range($\boldsymbol{W}$)
end procedure
```

Algorithm 7 Reduce the range of one basis matrix by the range of a second basis matrix

```
procedure ReduceRange($\boldsymbol{V}$, $\boldsymbol{W}$)
    $\boldsymbol{V} \in \mathbb{R}^{N \times n_v}$, $\boldsymbol{W} \in \mathbb{R}^{N \times n_w}$        ▷ Matrices which have to be combined
    $\epsilon \in \mathbb{R}_+$        ▷ Numerical tolerance, e.g. $\epsilon = 10^{-8}$
    $\boldsymbol{W} := \text{Orth}(\boldsymbol{W})$        ▷ Orthonormalise $\boldsymbol{W}$
    for $i := 1, 2, \ldots, n_v$ do        ▷ Repeat for all columns of $\boldsymbol{V}$
        $\boldsymbol{v} := \boldsymbol{V}_{:i} / |\boldsymbol{V}_{:i}|$        ▷ Column of $\boldsymbol{V}$. Normalisation is crucial in order to detect deflation
        $\boldsymbol{t} := \boldsymbol{W}^T \boldsymbol{v}$        ▷ Determine collinear parts of columns of $\boldsymbol{W}$ to $\boldsymbol{v}$
        $\boldsymbol{v} := \boldsymbol{v} - \boldsymbol{W}\boldsymbol{t}$        ▷ Eliminate parts of $\boldsymbol{v}$ which are collinear to $\boldsymbol{W}$
        if $|\boldsymbol{v}| > \epsilon$ then        ▷ Check for deflation
            if $\boldsymbol{Q}$ does not exist then
                $\boldsymbol{Q} := \boldsymbol{v} / |\boldsymbol{v}|$        ▷ Initialise output matrix
            else
                $\boldsymbol{Q} := \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{v}/|\boldsymbol{v}| \end{bmatrix}$        ▷ Normalise $\boldsymbol{v}$ and add to $\boldsymbol{Q}$
            end if
        end if
    end for
    $\boldsymbol{Q} := \text{Orth}(\boldsymbol{Q})$        ▷ Orthonormalise $\boldsymbol{Q}$
    return $\boldsymbol{Q}$        ▷ Orthonormal matrix with $\text{range}(\boldsymbol{Q}) = \text{range}(\boldsymbol{V}) - \text{range}(\boldsymbol{W})$
end procedure
```

expansion point in order to ensure that no eigenvalue is hit. In this thesis, in case of singular stiffness matrices, one expansion point is chosen at $s_e = 30\,\text{rad/s}$. This is well below the first eigenfrequency and far enough from the singular point to avoid numerical problems.

To match the FRFs at further frequencies, imaginary expansion points have to be used, since the FRF is the transfer function for $s = i\omega$. The expansion point σ_e, passed to *Algorithm* 3, is then the squared value of the imaginary expansion point, and thus, real negative.

In order to ensure that no expansion point matches any eigenvalue, the knowledge about the eigenvalues of the system gathered through *Algorithm* 4 can be used. An expansion point can then be placed between two known eigenvalues.

3.6 Real-World Example

Here, the KMS reduction results for a real-world structure are presented. In order to be able to calculate the frequency response of the original system for a sufficient number of frequencies, it is required to choose a system of moderate size. Therefore, the model used for the illustration is the Z-Axis from *Section* 3.1.5.2 with $N = 42\,921$ degrees of freedom, which is shown in *Fig.* 3.27. The structure is not bonded and has, thus, six rigid body modes. There are six interfaces regarded,

- one at each of the four linear guide carriages,
- one at the ball screw nut, and
- one at the tool centre point (TCP).

Each interface has six vectors, one for each Cartesian direction and one for the rotation around each Cartesian direction of the specific local coordinate system. This leads in total to $6 \times 6 = 36$ input vectors, which are output vectors at the same time. For more information, specific on modelling of interfaces, see *Chapter* 4.

The first 20 eigenfrequencies of this model are listed in *Tab.* 3.3. The first 11 modes are taken into account for the modal reduction. The first mode which is not explicitly matched is mode number 12 with an eigenfrequency of $6\,379\,\text{rad/s}$.

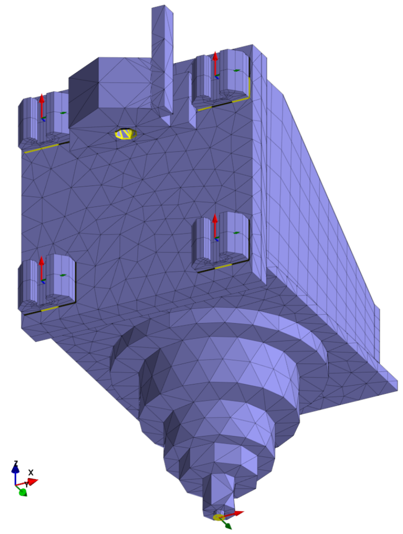


Figure 3.27: Z-Axis of the model from *Section* 3.1.5.2 with marked interfaces

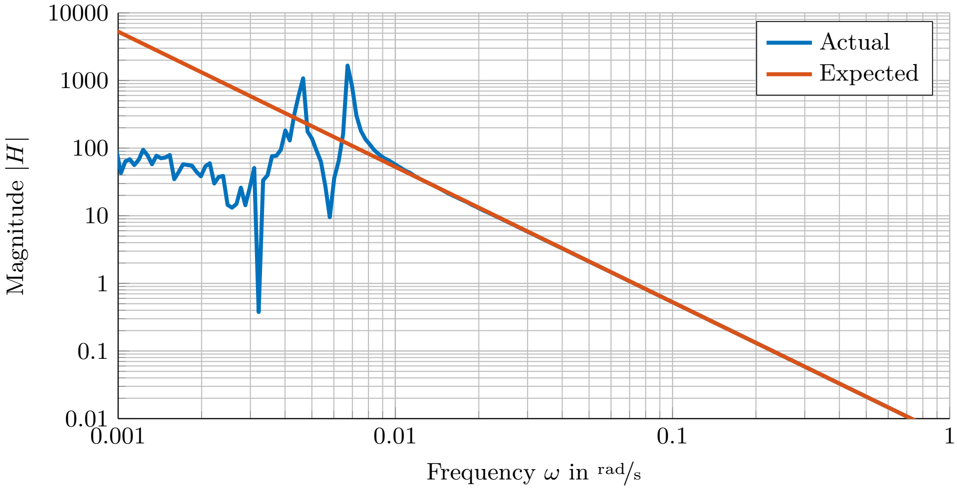


Figure 3.28: Detail of an example FRF of the original finite element model, which shows numerical issues during the calculation of the FRFs at low frequencies

The results for the relative error of all $36 \times 36 = 1\,296$ FRFs are shown in *Fig.* 3.30 to 3.32. The estimated error bounds show a very good accordance with the actual values, except for very low frequencies. The reason for the discrepancy at low frequencies can be found in numerical problems with the original model. As can be seen in *Tab.* 3.3, the first six eigenvalues are not exactly zero, what affects the FRFs of the original system. In *Fig.* 3.28, an FRF of the original system and the expected FRF are shown for a small section of the frequency range. A free body would exhibit a quadratic decay of the position magnitude with the frequency, which is a straight line in the logarithmic plot. The actual behaviour, however, is different and noisy for low frequencies. This is independent of the KMS method and appears in the same way for pure Krylov reduction, and thus, is not regarded as an issue for the KMS method.

At the transmission zeros, the relative error exceeds the estimated error bound locally in the same way as it was observed in *Section* 3.4.2.4. In *Fig.* 3.29, however, it is shown for the case of one expansion point that the relative error of the zeros is again, as predicted, below the estimated error bound.

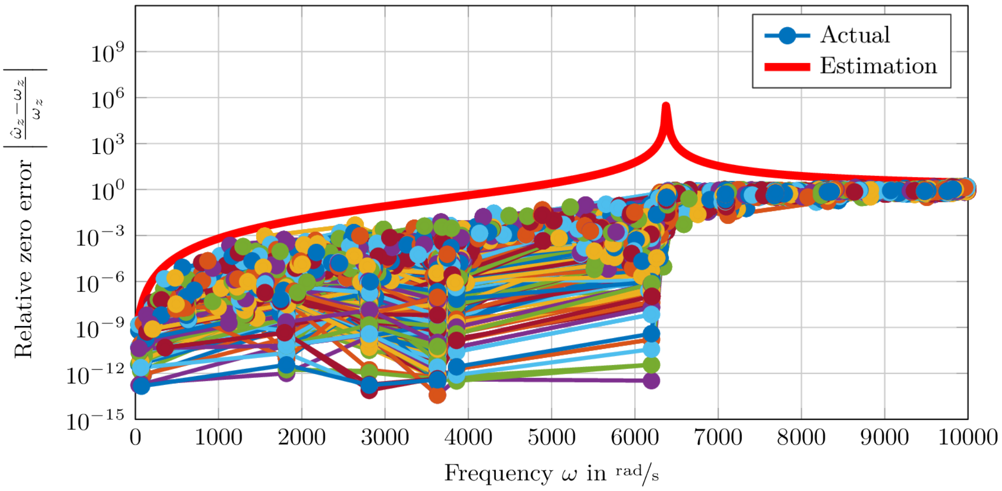


Figure 3.29: Relative error of all zero frequencies of the KMS based reduced real-world system's FRFs $(s_e = 30, m_e = 1, \omega_m = 6\,379)$

Table 3.3: First 20 eigenfrequencies of the original system from *Fig.* 3.27

k	**Eigenfrequency ω_k**	k	**Eigenfrequency ω_k**
1	0.0072	11	6 199
2	0.0045	12	6 379
3	0.0027	13	7 089
4	0.0026	14	7 687
5	0.0058	15	8 313
6	0.0086	16	8 356
7	1 816	17	9 480
8	2 809	18	9 750
9	3 629	19	9 821
10	3 858	20	10 318

3.7 Remarks on KMS Based Model Order Reduction

3.7.1 Analogy to Component Mode Synthesis

The special case of a KMS for a free mechanical structure (not bounded, six rigid body modes), with one expansion point near zero $s_e \approx 0$ and one Krylov iteration $m_e = 1$ corresponds to CMS with attachment modes and free interface normal modes as explained in *Section* 2.5.4.

If further, purely imaginary, expansion points are used, again with $m_e = 1$, this corresponds to adding frequency response modes to the CMS basis.

It thus follows that the error estimation from *Eq.* (3.145) can be used as error estimation for these special cases of CMS.

3.7.2 Application of KMS Based Reduction to Systems with First-Order Structure

Algorithm 1 is formulated specifically for systems with second-order structure, but all the subsequently called algorithms presented in *Section* 3.5.1 are formulated for systems with first-order structure according to *Eq.* (3.7). Here, with *Algorithm* 8, a procedure for the construction of a KMS basis for systems with first-order structure is added for completion.

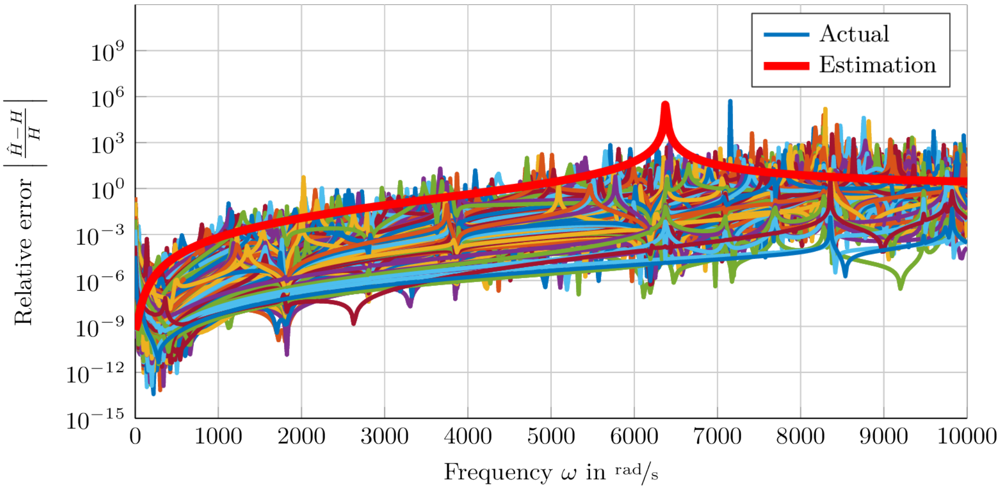


Figure 3.30: Relative error of all transfer functions of the real-world example after KMS based reduction with one Krylov expansion point $\left(s_e = 30,\, m_e = 1,\, \omega_m = 6\,379\right)$

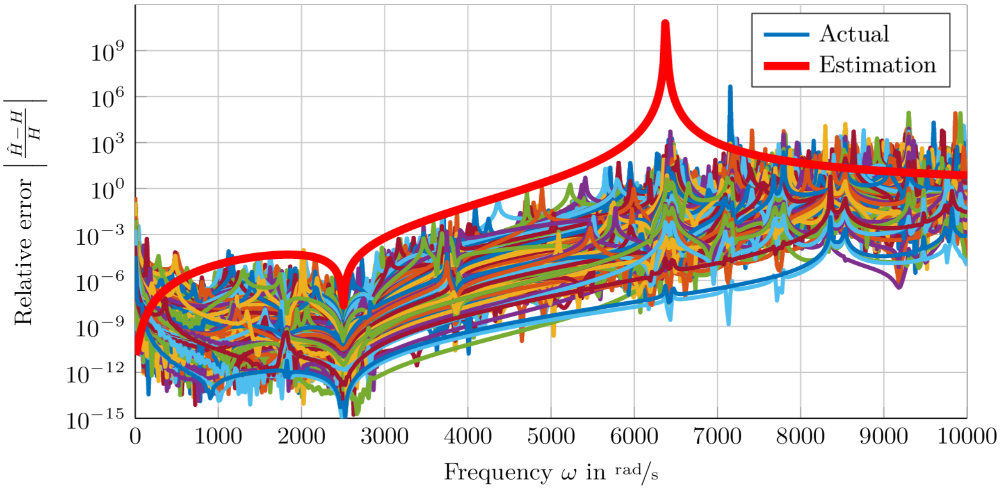


Figure 3.31: Relative error of all transfer functions of the real-world example after KMS based reduction with two Krylov expansion points $\left(\boldsymbol{s_e} = [30 \;\; i\,2\,500],\, \boldsymbol{m_e} = [1 \;\; 1],\, \omega_{mod} = 6\,379\right)$

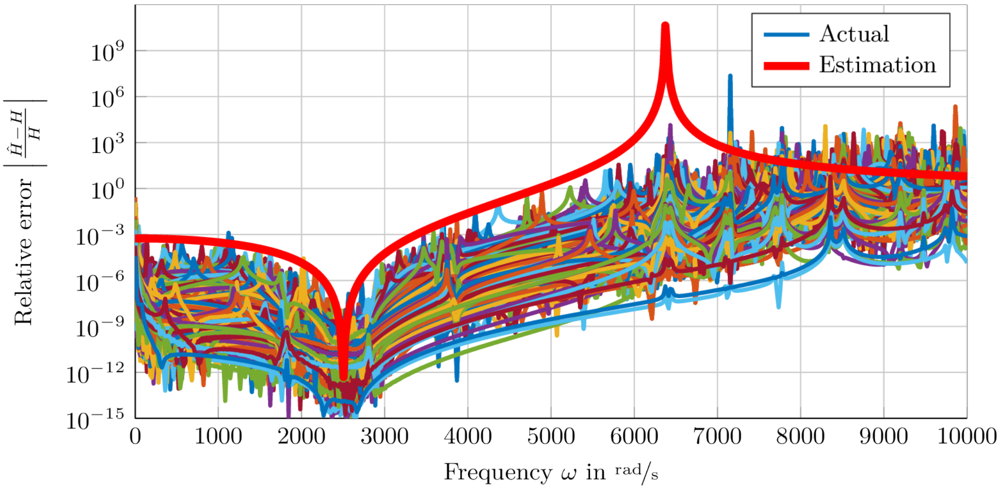


Figure 3.32: Relative error of all transfer functions of the real-world example after KMS based reduction with one Krylov expansion point and two Krylov iterations $\left(s_e = i\,2\,500,\, m_e = 2,\, \omega_m = 6\,379\right)$

Algorithm 8 KMS basis creation for systems with first-order structure

```
procedure FirstOrderKMS($A, E, B, C, \sigma_e, m_e, \omega_m$)
    $A, E \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times p}$, $C \in \mathbb{R}^{q\times N}$        ▷ System matrices in first-order structure
    $B_c$ := ExtendRange($B, C^T$)        ▷ Combine all inputs and outputs for the Krylov iteration
    $\sigma_e \in \mathbb{C}^{n_e}$        ▷ Expansion points
    $m_e \in \mathbb{Z}^{n_e}$        ▷ Number of Krylov iterations per expansion point
    $\omega_m \in \mathbb{R}_+$        ▷ Limit frequency for eigenvalues to keep
    $\Phi_M$ := Eigs($A, E, \omega_m$)        ▷ Solve eigenvalue problem
    $V_K$ := RationalArnoldi($A, E, B_c, m_e, \sigma_e$)        ▷ Create Krylov basis
    $V$ := ExtendRange($V_K, \Phi_M$)        ▷ Combine the modal and Krylov basis
    return $V$        ▷ Orthonormal KMS basis
end procedure
```

For a positive definite matrix $\boldsymbol{E}$ and a negative semi-definite matrix $\boldsymbol{A}$, the hypothesis for an upper bound of the relative error, analogous to *Eq.* (3.145),

$$|E_r(i\omega)| \leq \prod_{k=1}^{n_s} \left| \frac{\left(\omega_{e_k} + i\omega\right)^{2m_k}}{\left(\omega_0 + i\omega\right)^{2m_k}} \right| = \hat{E}_r(i\omega)\ . \tag{3.152}$$

is evaluated. Because of the definiteness of the system matrices, the eigenvalues are all real-valued or zero. Therefore, the FRFs do not have any poles and, usually, also no zeros, what leads to a smooth relative error.

The relative error for a set of 20 random systems, generated equally as presented in *Section* 3.1.5.1, but with the system matrices

$$\boldsymbol{E} = \boldsymbol{I}\ , \quad \boldsymbol{A} = -\mathrm{diag}(\omega_1, \omega_2, \ldots, \omega_N)\ , \tag{3.153}$$

is shown in *Fig.* 3.33 for a KMS based reduction with a single expansion point. This shows that the idea of KMS based reduction can be expanded to systems of first-order structure. However, more work is required in order to fully analyse the characteristics of the method.

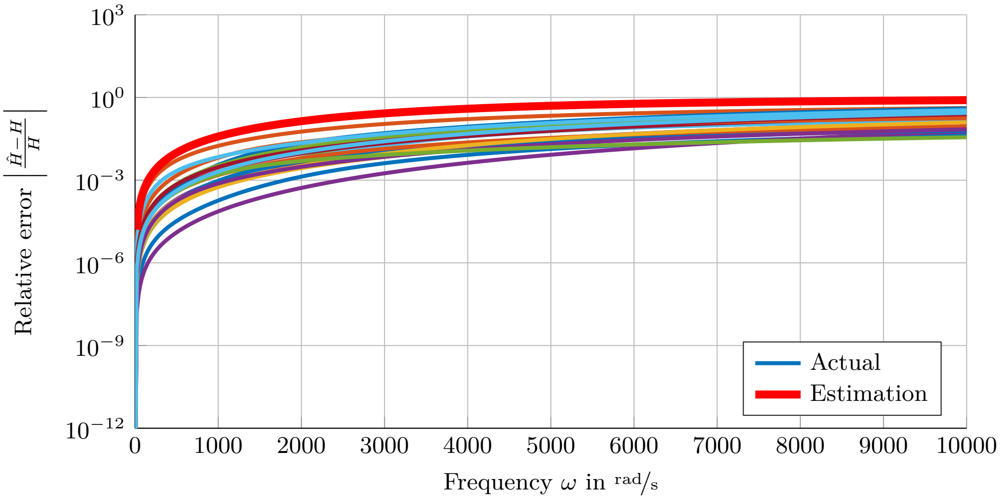


Figure 3.33: Relative error of the FRFs of 20 randomly generated systems with first-order structure after KMS based reduction with one Krylov expansion point $\left(\sigma_e = 0, m_e = 1, \omega_m = 5\,000\right)$

An example for models with first-order structure are thermodynamic systems according to

$$\boldsymbol{C_T}\dot{\boldsymbol{x}}_{\boldsymbol{T}} + \boldsymbol{K_T}\boldsymbol{x_T} = \boldsymbol{B_T}\boldsymbol{u_T} \tag{3.154}$$

with the thermal capacity matrix $\boldsymbol{C_T}$, the conductivity matrix $\boldsymbol{K_T}$, the temperature state vector $\boldsymbol{x_T}$, the thermal input matrix $\boldsymbol{B_T}$, and the heat flow input vector $\boldsymbol{u_T}$. This type of systems could be handled by calling *Algorithm* 8 with the system matrices

$$\boldsymbol{A} = -\boldsymbol{K_T} \text{ and } \quad \boldsymbol{E} = \boldsymbol{C_T} \tag{3.155}$$

in order to get the appropriate KMS basis for projection.

4 Interfaces to Flexible Bodies

Interfaces to flexible bodies are either force / torque inputs or displacement / rotation outputs and can exhibit six degrees of freedom (DOFs) in total, three translations and three rotations. Usually, the interface is acting on a distinct surface of the body. In order to enable an efficient work-flow, no force distributing elements of the commercial finite element software should be used, because these have to be defined in an early stage of the modelling, usually before the user is aware of every interface required for coupling and control of the structure. Changing the interfaces in a later stage of modelling would then require a high effort, because the original finite element model would have to be modified again. In *Section* 4.1, the distribution of loads onto finite element meshed surfaces is introduced and used in *Section* 4.2 in order to model stationary interfaces with six DOFs.

In order to model moving machine parts like, e.g. linear guides, moving interfaces have to be modelled. Because the order of a reduced system depends on the number of interfaces to the system for all accurate model order reduction methods including KMS based reduction, it is not applicable to interact with every node of an interface surface independently. Therefore, in *Section* 4.3, a method for the approximation of moving interfaces using trigonometric interpolation is presented and the methods are discussed in *Section* 4.4.

4.1 Distributed Loads on Finite Element Surfaces

Interfacing with finite element models requires the formulation of distributed loads and displacements in terms of nodal loads and displacements. An example of a distributed load q that depends on arbitrary local coordinates (u, v, w) is shown in *Fig.* 4.1 with the highlighted region associated with a hexahedral element e_k.

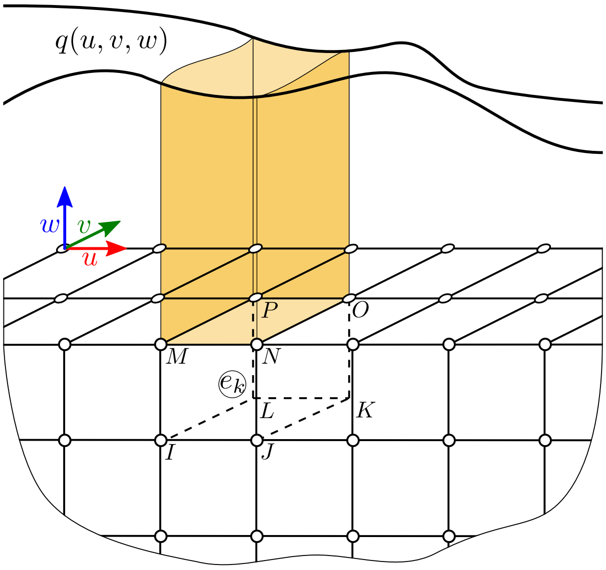


Figure 4.1: Distributed load on a finite element surface

The derivation of element consistent nodal forces requires the introduction of external work functionals in the variational formulation of the finite element stiffness equation, as shown e.g. by Felippa

[36]. This also involves knowledge about the element shape functions. Because these element shape functions are not always accessible for all types of elements provided by a commercial finite element modelling software, an alternative approach is used here, which involves direct lumping of distributed loads similar to the element by element lumping method described by Felippa [36].

First, the distributed load is integrated over each involved element surface in order to receive a lumped load with the same resulting value and centre of action as the distributed load. Subsequently, the lumped load is distributed to the corner nodes of the element. The advantage of this method is that it only involves knowledge about the geometry of the elements and not the element shape functions.

4.1.1 Resulting Element Surface Load

The load $\gamma^{(k)}$ resulting from a distributed load q on a surface $A^{(k)}$ of the element e_k is calculated by integration over the surface according to

$$\gamma^{(k)} = \iint\limits_{A^{(k)}} q(u, v, w)\, dA \ . \tag{4.1}$$

The superscript $^{(k)}$ with round brackets denotes the assignment to the element number k. The centre of action $\boldsymbol{r}_{\boldsymbol{C}}^{(k)}$ can be calculated using the u-, v-, and w-weighted integrals

$$\gamma_u^{(k)} = \iint\limits_{A^{(k)}} u \cdot q(u, v, w)\, dA \ , \quad \gamma_v^{(k)} = \iint\limits_{A^{(k)}} v \cdot q(u, v, w)\, dA \ , \quad \gamma_w^{(k)} = \iint\limits_{A^{(k)}} w \cdot q(u, v, w)\, dA \ , \tag{4.2}$$

according to

$$\boldsymbol{r}_{\boldsymbol{C}}^{(k)} = \begin{bmatrix} \frac{\gamma_u^{(k)}}{\gamma^{(k)}} & \frac{\gamma_v^{(k)}}{\gamma^{(k)}} & \frac{\gamma_w^{(k)}}{\gamma^{(k)}} \end{bmatrix}^T \ . \tag{4.3}$$

In order to enable efficient evaluation of the integrals from *Eq.* (4.1) and (4.2), the integration can be approximated by means of Gaussian quadrature formulas as listed, e.g. by Cowper [24].

For the case that the integration limits do not coincide with the boundaries of the finite element surfaces, e.g. if the interface surface is only a subregion of a topological surface of the model, these element surfaces can be split into multiple triangles and then be handled likewise.

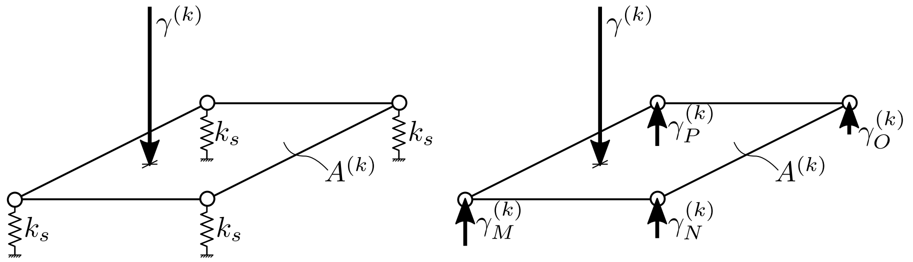
4.1.2 Quadrilateral Element Surfaces

Finite element surface meshes usually consist of quadrilateral and triangular elements. For a quadrilateral element surface, the total load $\gamma^{(k)}$ on an element has to be distributed on the four corner nodes, what is not a statically determined load case. Therefore, the element is handled as an elastically mounted plate with equal stiffness k_s at each corner node according to *Fig.* 4.2(a). The force reactions at the springs are then used as nodal forces, as outlined in *Fig.* 4.2(b).

First, a local coordinate system $\left(\xi^{(k)}, \eta^{(k)}, \zeta^{(k)}\right)$ is defined, where the $\xi^{(k)}$-$\eta^{(k)}$-plane lies in the plane of the element surface as shown in *Fig.* 4.3. The coordinate system can be defined using the corner nodes M, N, and P according to

$$\boldsymbol{e}_{\boldsymbol{\xi}}^{(k)} = \frac{\boldsymbol{r}_{\boldsymbol{N}}^{(k)} - \boldsymbol{r}_{\boldsymbol{M}}^{(k)}}{\left|\boldsymbol{r}_{\boldsymbol{N}}^{(k)} - \boldsymbol{r}_{\boldsymbol{M}}^{(k)}\right|} \ , \quad \bar{\boldsymbol{e}}_{\boldsymbol{\eta}}^{(k)} = \frac{\boldsymbol{r}_{\boldsymbol{P}}^{(k)} - \boldsymbol{r}_{\boldsymbol{M}}^{(k)}}{\left|\boldsymbol{r}_{\boldsymbol{P}}^{(k)} - \boldsymbol{r}_{\boldsymbol{M}}^{(k)}\right|} \ , \quad \boldsymbol{e}_{\boldsymbol{\zeta}}^{(k)} = \boldsymbol{e}_{\boldsymbol{\xi}}^{(k)} \times \bar{\boldsymbol{e}}_{\boldsymbol{\eta}}^{(k)} \ , \quad \boldsymbol{e}_{\boldsymbol{\eta}}^{(k)} = \boldsymbol{e}_{\boldsymbol{\zeta}}^{(k)} \times \boldsymbol{e}_{\boldsymbol{\xi}}^{(k)} \ . \tag{4.4}$$

This leads to an axis $\boldsymbol{e}_{\boldsymbol{\xi}}^{(k)}$ that is parallel to the M-N-edge, an axis $\boldsymbol{e}_{\boldsymbol{\zeta}}^{(k)}$ that is normal to the element surface, and an axis $\boldsymbol{e}_{\boldsymbol{\eta}}^{(k)}$ that is normal to both, $\boldsymbol{e}_{\boldsymbol{\xi}}^{(k)}$ and $\boldsymbol{e}_{\boldsymbol{\zeta}}^{(k)}$.



(a) Model used for load distribution (b) Free body diagram and force reactions

Figure 4.2: Load distribution on a quadrilateral finite element surface

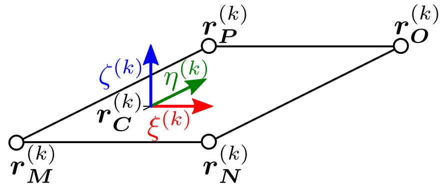


Figure 4.3: Element coordinate system and nodal position vectors

The $\xi^{(k)}$- and $\eta^{(k)}$-coordinates for all four corner points relative to the centroid $\boldsymbol{r}_{\boldsymbol{C}}^{(k)}$ are then found by projection according to

$$\begin{aligned}
{\xi_M}^{(k)} &= \boldsymbol{e}_{\boldsymbol{\xi}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{M}}^{(k)} - \boldsymbol{r}_{\boldsymbol{C}}^{(k)}\right) \ , & {\eta_M}^{(k)} &= \boldsymbol{e}_{\boldsymbol{\eta}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{M}}^{(k)} - \boldsymbol{r}_{\boldsymbol{C}}^{(k)}\right) \ , \\
{\xi_N}^{(k)} &= \boldsymbol{e}_{\boldsymbol{\xi}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{N}}^{(k)} - \boldsymbol{r}_{\boldsymbol{C}}^{(k)}\right) \ , & {\eta_N}^{(k)} &= \boldsymbol{e}_{\boldsymbol{\eta}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{N}}^{(k)} - \boldsymbol{r}_{\boldsymbol{C}}^{(k)}\right) \ , \\
{\xi_O}^{(k)} &= \boldsymbol{e}_{\boldsymbol{\xi}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{O}}^{(k)} - \boldsymbol{r}_{\boldsymbol{C}}^{(k)}\right) \ , & {\eta_O}^{(k)} &= \boldsymbol{e}_{\boldsymbol{\eta}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{O}}^{(k)} - \boldsymbol{r}_{\boldsymbol{C}}^{(k)}\right) \ , \\
{\xi_P}^{(k)} &= \boldsymbol{e}_{\boldsymbol{\xi}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{P}}^{(k)} - \boldsymbol{r}_{\boldsymbol{C}}^{(k)}\right) \ , & {\eta_P}^{(k)} &= \boldsymbol{e}_{\boldsymbol{\eta}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{P}}^{(k)} - \boldsymbol{r}_{\boldsymbol{C}}^{(k)}\right) \ .
\end{aligned} \tag{4.5}$$

For the calculation of the four one-dimensional reaction forces, four equations have to be derived. The first equation results from the force equilibrium according to

$$\gamma_M^{(k)} + \gamma_N^{(k)} + \gamma_O^{(k)} + \gamma_P^{(k)} = \gamma^{(k)} \ . \tag{4.6}$$

Two more equations result from the conditions of equilibrium for torques around $\xi^{(k)}$ and $\eta^{(k)}$ according to

$$\xi_M^{(k)}\,\gamma_M^{(k)} + \xi_N^{(k)}\,\gamma_N^{(k)} + \xi_O^{(k)}\,\gamma_O^{(k)} + \xi_P^{(k)}\,\gamma_P^{(k)} = 0 \tag{4.7}$$

$$\eta_M^{(k)}\,\gamma_M^{(k)} + \eta_N^{(k)}\,\gamma_N^{(k)} + \eta_O^{(k)}\,\gamma_O^{(k)} + \eta_P^{(k)}\,\gamma_P^{(k)} = 0 \ . \tag{4.8}$$

One more equation can be derived by constraining the elastically mounted plate to move rigidly, i.e. in a plane. This leads to the equations

$$\begin{aligned}
\frac{1}{k_s}\,\gamma_O^{(k)} &= \frac{1}{k_s}\,\gamma_M^{(k)} + \frac{\lambda_N^{(k)}}{k_s}\left(\gamma_N^{(k)} - \gamma_M^{(k)}\right) + \frac{\lambda_P^{(k)}}{k_s}\left(\gamma_P^{(k)} - \gamma_M^{(k)}\right) \\
\gamma_O^{(k)} &= \gamma_M^{(k)} + \lambda_N^{(k)}\left(\gamma_N^{(k)} - \gamma_M^{(k)}\right) + \lambda_P^{(k)}\left(\gamma_P^{(k)} - \gamma_M^{(k)}\right)
\end{aligned} \tag{4.9}$$

with the geometrical relations between the corner points

$$\begin{bmatrix} \lambda_N^{(k)} \\ \lambda_P^{(k)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_{\boldsymbol{\xi}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{N}}^{(k)} - \boldsymbol{r}_{\boldsymbol{M}}^{(k)}\right) & \boldsymbol{e}_{\boldsymbol{\xi}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{P}}^{(k)} - \boldsymbol{r}_{\boldsymbol{M}}^{(k)}\right) \\ \boldsymbol{e}_{\boldsymbol{\eta}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{N}}^{(k)} - \boldsymbol{r}_{\boldsymbol{M}}^{(k)}\right) & \boldsymbol{e}_{\boldsymbol{\eta}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{P}}^{(k)} - \boldsymbol{r}_{\boldsymbol{M}}^{(k)}\right) \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{e}_{\boldsymbol{\xi}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{O}}^{(k)} - \boldsymbol{r}_{\boldsymbol{M}}^{(k)}\right) \\ \boldsymbol{e}_{\boldsymbol{\eta}}^{(k)} \bullet \left(\boldsymbol{r}_{\boldsymbol{O}}^{(k)} - \boldsymbol{r}_{\boldsymbol{M}}^{(k)}\right) \end{bmatrix} \ . \tag{4.10}$$

Finally, with *Eq.* (4.6), (4.7), (4.8), and (4.9) four conditions on the nodal forces are defined, which can be formulated as a system of equations according to

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ \xi_M^{(k)} & \xi_N^{(k)} & \xi_O^{(k)} & \xi_P^{(k)} \\ \eta_M^{(k)} & \eta_N^{(k)} & \eta_O^{(k)} & \eta_P^{(k)} \\ 1 - \lambda_N^{(k)} - \lambda_N^{(k)} & \lambda_N^{(k)} & \lambda_P^{(k)} & -1 \end{bmatrix} \begin{bmatrix} \gamma_M^{(k)} \\ \gamma_N^{(k)} \\ \gamma_O^{(k)} \\ \gamma_P^{(k)} \end{bmatrix} = \begin{bmatrix} \gamma^{(k)} \\ 0 \\ 0 \\ 0 \end{bmatrix} . \tag{4.11}$$

4.1.3 Triangular Element Surfaces

Triangular element surfaces demand three equations for the distribution of the total load on three nodes according to *Fig.* 4.4. Analogous to quadrilateral surfaces, the force equilibrium

$$\gamma_M^{(k)} + \gamma_N^{(k)} + \gamma_P^{(k)} = \gamma^{(k)} \tag{4.12}$$

provides one equation and the torque equilibria

$$\begin{aligned} \xi_M^{(k)} \gamma_M^{(k)} + \xi_N^{(k)} \gamma_N^{(k)} + \xi_P^{(k)} \gamma_P^{(k)} &= 0 \\ \eta_M^{(k)} \gamma_M^{(k)} + \eta_N^{(k)} \gamma_N^{(k)} + \eta_P^{(k)} \gamma_P^{(k)} &= 0 \end{aligned} \tag{4.13}$$

provide two more equations. The local coordinates of the points are again obtained with *Eq.* (4.5). The resulting system of equations reads

$$\begin{bmatrix} 1 & 1 & 1 \\ \xi_M^{(k)} & \xi_N^{(k)} & \xi_P^{(k)} \\ \eta_M^{(k)} & \eta_N^{(k)} & \eta_P^{(k)} \end{bmatrix} \begin{bmatrix} \gamma_M^{(k)} \\ \gamma_N^{(k)} \\ \gamma_P^{(k)} \end{bmatrix} = \begin{bmatrix} \gamma^{(k)} \\ 0 \\ 0 \end{bmatrix} . \tag{4.14}$$

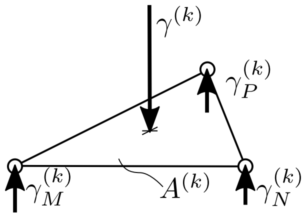


Figure 4.4: Load distribution on a triangular finite element surface

4.1.4 Assembly of Interface Vectors

So far, nodal weights corresponding to distributed loads on finite element surfaces were derived on an element-basis. The nodal weight $\gamma_i^{(k)}$ for the node i of element k is scalar and can be interpreted as a weighting factor for any nodal degree of freedom or for any multidimensional load, i.e. any vectorial load. Because, in general, multiple elements are connected to the same node, the total nodal weight

$\gamma^{[i]}$ for the node i of the finite element mesh have to be summed up from all contributing element node weights according to

$$\gamma^{[i]} = \sum_{k \in \mathscr{E}} \sum_{l \in \mathscr{N}^{(k)}} \gamma_l^{(k)} \, \delta_l^{(k)}(i) \tag{4.15}$$

for each node in the set $\mathscr{N}^{(k)}$ of every element in the set $\mathscr{E}$, where the contribution factor

$$\delta_l^{(k)}(i) = \begin{cases} 1 & \text{if node } l \text{ of element } k \text{ equals the node } i \text{ of the mesh} \\ 0 & \text{else} \end{cases} \tag{4.16}$$

is used to account for the element connectivity.

In order to use the nodal loads with the dynamic system according to

$$\begin{aligned} \boldsymbol{M}\,\ddot{\boldsymbol{x}} + \boldsymbol{D}\,\dot{\boldsymbol{x}} + \boldsymbol{K}\,\boldsymbol{x} &= \boldsymbol{b}\,u \\ \boldsymbol{y} &= \boldsymbol{c}\,\boldsymbol{x}\,, \end{aligned} \tag{4.17}$$

the input vector $\boldsymbol{b}$ has to be assembled appropriately. The input and output vectors of a system can be handled equivalently, and therefore only the inputs are discussed in the following and denoted as interface vectors. An interface vector which is used as output vector leads to a weighted summation of the nodal degrees of freedom, i.e. a weighted averaging of the nodal displacements.

The state vector $\boldsymbol{x}$ of a finite element system contains the nodal coordinates of the system in an arbitrary order defined during the assembly of the finite element matrices. For each degree of freedom d of a node i, an allocation vector $\boldsymbol{p}_d^{[i]} \in \mathbb{R}^N$ can be defined. This is an all zero vector with the exception of an entry of 1 at the place of the corresponding degree of freedom. Assembling the interface vector $\boldsymbol{b_d}$ is then achieved by summing up the contributions of the nodal weights $\gamma^{[i]}$ for all nodes according to

$$\boldsymbol{b_d} = \sum_{i=1}^{N_n} \boldsymbol{p}_d^{[i]} \, \gamma^{[i]}\,, \tag{4.18}$$

where the index d stands for the degree of freedom of the nodes which shall be addressed, e.g. the Cartesian displacements of the node in the global coordinates x, y, and z.

4.2 Stationary Interface Modelling

Interfaces which do not change the position of action on a flexible body, i.e. the set of finite element nodes affected does not change, are called stationary interfaces. They are used for modelling of, e.g. machine support elements, bearing interfaces, process forces on a tool, or readout of the tool centre point (TCP) position and orientation.

4.2.1 Requirements on Stationary Interface Models

In general, the interfaces are used to interact with all six spatial degrees of freedom, three Cartesian translations and three rotations in any local coordinate system (u, v, w) and $(\varphi_u, \varphi_v, \varphi_w)$, respectively. Either forces and torques are to be applied to the structure, or displacements and rotations are to be evaluated. The centre of action of the interface has to lie at a specifiable location $\boldsymbol{r_0}$. The situation is shown in *Fig.* 4.5.

Through the cantilever between the center of area of surface A and the resulting location of action $\boldsymbol{r_0}$, a force in, e.g. u-direction must introduce a torque around the v-axis on the surface in the shown example. The formalism for the definition of stationary interfaces has thus to account for the location of

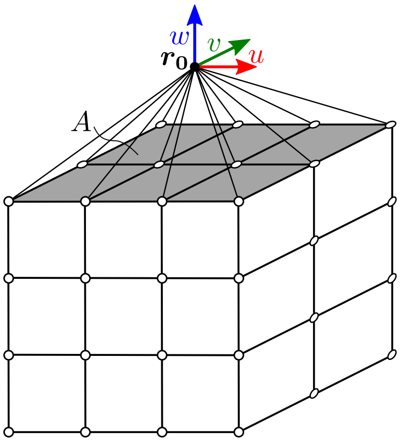


Figure 4.5: Example of a stationary interface to the top surface of a cube and figurative connection of the interface reference point to the finite element nodes

action. Moreover, the interface's degrees of freedom have to be orthonormal, i.e. an input of magnitude one has to result in a total force or torque with magnitude one in the specified direction and zero in all other directions.

The forces have to be distributed in a meaningful manner to the finite element nodes of the specified surface and the surface should not be stiffened or even rigidified. This specification results from comparisons between simulations and experiments carried out by Maglie [71].

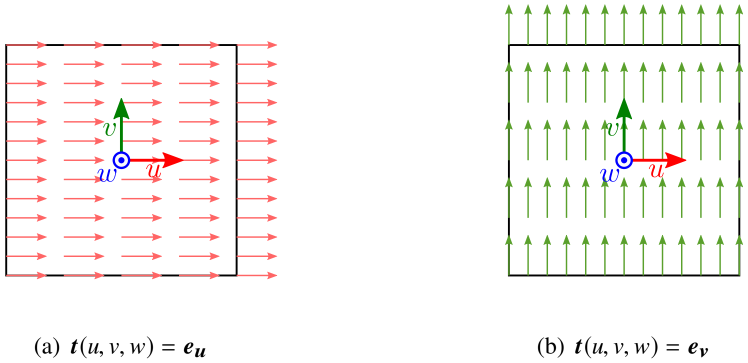
4.2.2 Elementary Force Distributions

In a first step, elementary force distributions are specified, which allow the application of loads in all six DOFs. The resulting total interface loads are then analysed and used for orthonormalisation of the interface's DOFs in a second step. It is assumed that all nodes on the surface of interest exhibit three structural DOFs in global coordinates (x, y, z).

Table 4.1: Elementary force distributions

DOF	Description	Vector field $\boldsymbol{t}(u, v, w)$	Nodal DOF vectors
u	Translation in u-direction	$\boldsymbol{e_u}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{u}} = \boldsymbol{e_u}\,\gamma^{[i]}$
v	Translation in v-direction	$\boldsymbol{e_v}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{v}} = \boldsymbol{e_v}\,\gamma^{[i]}$
w	Translation in w-direction	$\boldsymbol{e_w}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{w}} = \boldsymbol{e_w}\,\gamma^{[i]}$
φ_u	Rotation around u	$v\,\boldsymbol{e_w} - w\,\boldsymbol{e_v}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{\varphi_u}} = \boldsymbol{e_w}\,\gamma_v^{[i]} - \boldsymbol{e_v}\,\gamma_w^{[i]}$
φ_v	Rotation around v	$w\,\boldsymbol{e_u} - u\,\boldsymbol{e_w}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{\varphi_v}} = \boldsymbol{e_u}\,\gamma_w^{[i]} - \boldsymbol{e_w}\,\gamma_u^{[i]}$
φ_w	Rotation around w	$u\,\boldsymbol{e_v} - v\,\boldsymbol{e_u}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{\varphi_w}} = \boldsymbol{e_v}\,\gamma_u^{[i]} - \boldsymbol{e_u}\,\gamma_v^{[i]}$

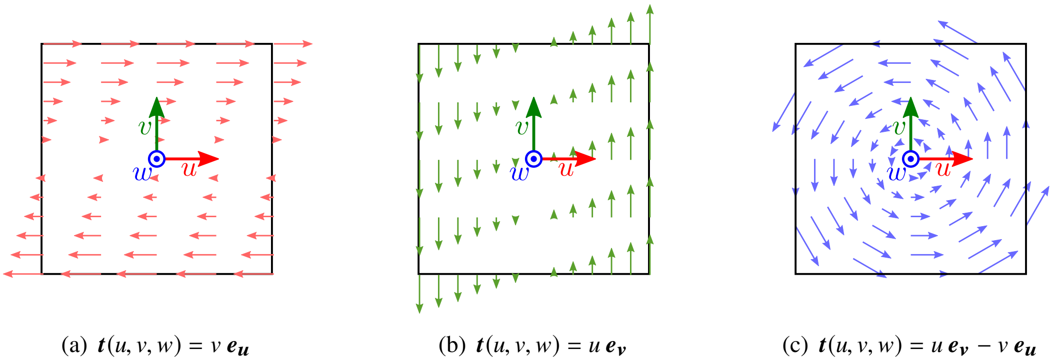
The elementary force distributions are composed of vector fields of the form $q(u, v, w)\,\boldsymbol{t_c}$ with constant direction $\boldsymbol{t_c}$ and position dependent value q. The definitions of the vector fields used here are listed in *Tab.* 4.1. The vectors $\boldsymbol{e_u}$, $\boldsymbol{e_v}$, and $\boldsymbol{e_w}$ denote the unit vectors in direction of u, v, and w of the local coordinate system. In order to introduce translational degrees of freedom, a constant vector field with the desired direction is used. The vector fields for the top-view of the example from *Fig.* 4.5 are shown in *Fig.* 4.6.



(a) $\boldsymbol{t}(u, v, w) = \boldsymbol{e_u}$ (b) $\boldsymbol{t}(u, v, w) = \boldsymbol{e_v}$

Figure 4.6: Force-generating vector fields for the planar degrees of freedom

The force distribution for the introduction of rotational degrees of freedom is inspired from the stress distribution in a cross section of a beam under bending load. The stress value increases linearly with the distance to the neutral line, or here, with the distance to the origin of the local coordinate system. There are two combinations leading to the same torque each, which are shown in *Fig.* 4.7(a) and *Fig.* 4.7(b). Combining both leads to tangential vectors concentric around the origin of the local coordinate system, as shown in *Fig.* 4.7(c).



(a) $\boldsymbol{t}(u, v, w) = v\, \boldsymbol{e_u}$ (b) $\boldsymbol{t}(u, v, w) = u\, \boldsymbol{e_v}$ (c) $\boldsymbol{t}(u, v, w) = u\, \boldsymbol{e_v} - v\, \boldsymbol{e_u}$

Figure 4.7: Torque-generating vector fields for the planar degrees of freedom

Four different weighting functions q are required in order to build all vector fields from *Tab.* 4.1, these are $q = 1$ for the constant value and $q = u$, $q = v$, and $q = w$ for the weighted vector fields. These weighting functions are used to calculate the nodal weights $\gamma^{[i]}$, $\gamma_u^{[i]}$, $\gamma_v^{[i]}$, and $\gamma_w^{[i]}$, respectively, with the procedure described in *Section* 4.1. This leads to the nodal DOF vectors $\bar{\boldsymbol{f}}_{\boldsymbol{a}}^{[i]}$ according to the last column of *Tab.* 4.1, whereby the index a stands for one of the six degrees of freedom of the interface. The bar on the $\bar{\boldsymbol{f}}$ denotes that these are the elementary and not the final nodal DOF vectors, because they are not yet orthonormalised.

4.2.3 Orthonormal Interfaces

The force $\bar{F}_{R_{a,b}}$ in direction a, resulting from a set of nodal DOF vectors $\bar{\boldsymbol{f}}_{\boldsymbol{b}}^{[i]}$ is calculated through summation of the projected nodal DOF vectors onto the unit vector in the desired direction according to

$$\bar{F}_{R_{a,b}} = \sum_{i=1}^{N_n} \boldsymbol{e_a} \bullet \bar{\boldsymbol{f}}_{\boldsymbol{b}}^{[i]} \qquad a \in \{u, v, w\} \ , \quad b \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\} \quad (4.19)$$

over all N_n nodes. Furthermore, the resulting torques $\bar{M}_{R_{a,b}}$ induced by the nodal DOF vectors related to the reference point $\boldsymbol{r_0}$ are the sum of the torques induced by the all single nodes, projected onto the unit vector of the desired direction, according to

$$\bar{M}_{R_{a,b}} = \sum_{i=1}^{N_n} \boldsymbol{e_a} \bullet \left(\left(\boldsymbol{r}^{[i]} - \boldsymbol{r_0} \right) \times \bar{\boldsymbol{f}}_{\boldsymbol{b}}^{[i]} \right) \qquad a \in \{u, v, w\} \ , \quad b \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\} \ . \tag{4.20}$$

The resulting forces and torques can be combined to a resulting load matrix $\bar{\boldsymbol{F}}_{\boldsymbol{R}}$ of dimension 6×6 as

$$\bar{\boldsymbol{F}}_{\boldsymbol{R}} = \begin{bmatrix} \bar{F}_{R_{u,u}} & \bar{F}_{R_{u,v}} & \bar{F}_{R_{u,w}} & \bar{F}_{R_{u,\varphi_u}} & \bar{F}_{R_{u,\varphi_v}} & \bar{F}_{R_{u,\varphi_w}} \\ \bar{F}_{R_{v,u}} & \cdots & \cdots & \cdots & \cdots & \bar{F}_{R_{v,\varphi_w}} \\ \bar{F}_{R_{w,u}} & \cdots & \cdots & \cdots & \cdots & \bar{F}_{R_{w,\varphi_w}} \\ \bar{M}_{R_{u,u}} & \bar{M}_{R_{u,v}} & \bar{M}_{R_{u,w}} & \bar{M}_{R_{u,\varphi_u}} & \bar{M}_{R_{u,\varphi_v}} & \bar{M}_{R_{u,\varphi_w}} \\ \bar{M}_{R_{v,u}} & \cdots & \cdots & \cdots & \cdots & \bar{M}_{R_{v,\varphi_w}} \\ \bar{M}_{R_{w,u}} & \cdots & \cdots & \cdots & \cdots & \bar{M}_{R_{w,\varphi_w}} \end{bmatrix} . \tag{4.21}$$

This matrix is invertible because the elementary force distributions allow the application of all six degrees of freedom and its inverse is defined as the orthonormalisation matrix $\boldsymbol{R_F}$ according to

$$\boldsymbol{R_F} = \begin{bmatrix} R_{F_{u,u}} & R_{F_{u,v}} & R_{F_{u,w}} & R_{F_{u,\varphi_u}} & R_{F_{u,\varphi_v}} & R_{F_{u,\varphi_w}} \\ R_{F_{v,u}} & \ddots & \vdots & \vdots & \vdots & \vdots \\ R_{F_{w,u}} & \cdots & \ddots & \vdots & \vdots & \vdots \\ R_{F_{\varphi_u,u}} & \cdots & \cdots & \ddots & \vdots & \vdots \\ R_{F_{\varphi_v,u}} & \cdots & \cdots & \cdots & \ddots & \vdots \\ R_{F_{\varphi_w,u}} & \cdots & \cdots & \cdots & \cdots & R_{F_{\varphi_w,\varphi_w}} \end{bmatrix} = \bar{\boldsymbol{F}}_{\boldsymbol{R}}^{-1} \tag{4.22}$$

This matrix is then used in order to find an orthonormal set of nodal DOF vectors $\boldsymbol{f}_{\boldsymbol{a}}^{[i]}$ according to

$$\boldsymbol{f}_{\boldsymbol{a}}^{[i]} = \sum_{b} R_{F_{a,b}} \bar{\boldsymbol{f}}_{\boldsymbol{b}}^{[i]} \qquad a, b \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\} \ , \tag{4.23}$$

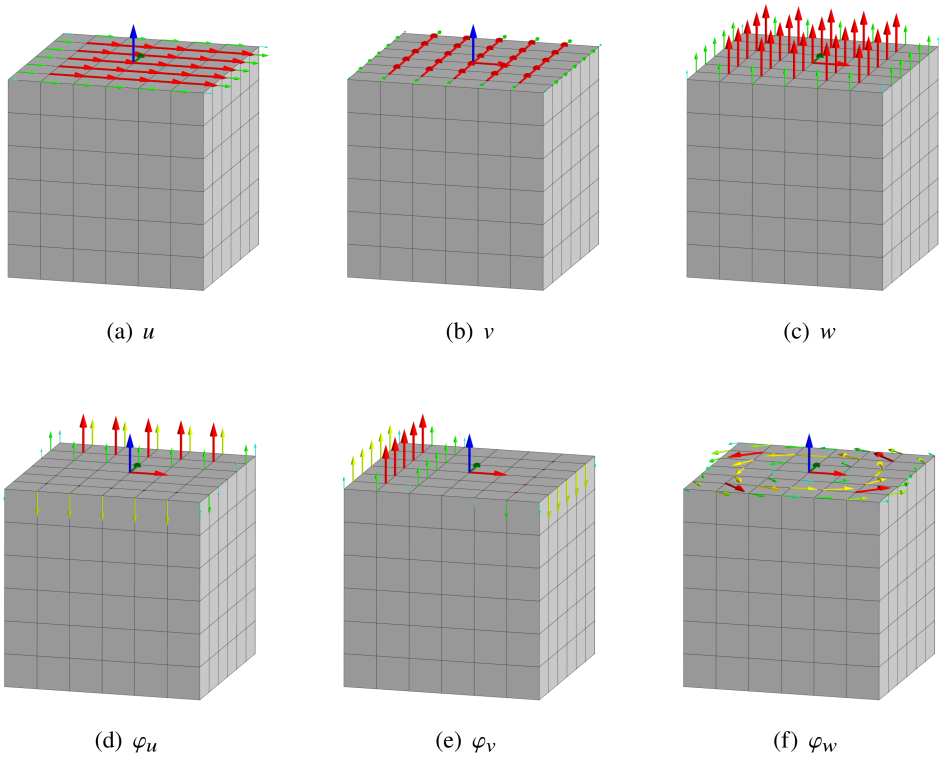
what mimes a matrix multiplication of the elementary force distributions with the orthonormalisation matrix.

The resulting load matrix $\boldsymbol{F_R}$ for the normalised nodal DOF vectors, determined analogously as in *Eq.* (4.21), but with the normalised nodal DOF vectors $\boldsymbol{f}_{\boldsymbol{a}}^{[i]}$, is then the identity matrix and the interface is called orthonormal.

4.2.4 Examples

In order to visualise the nodal load distribution, three examples are shown by means of a finite element meshed cube.

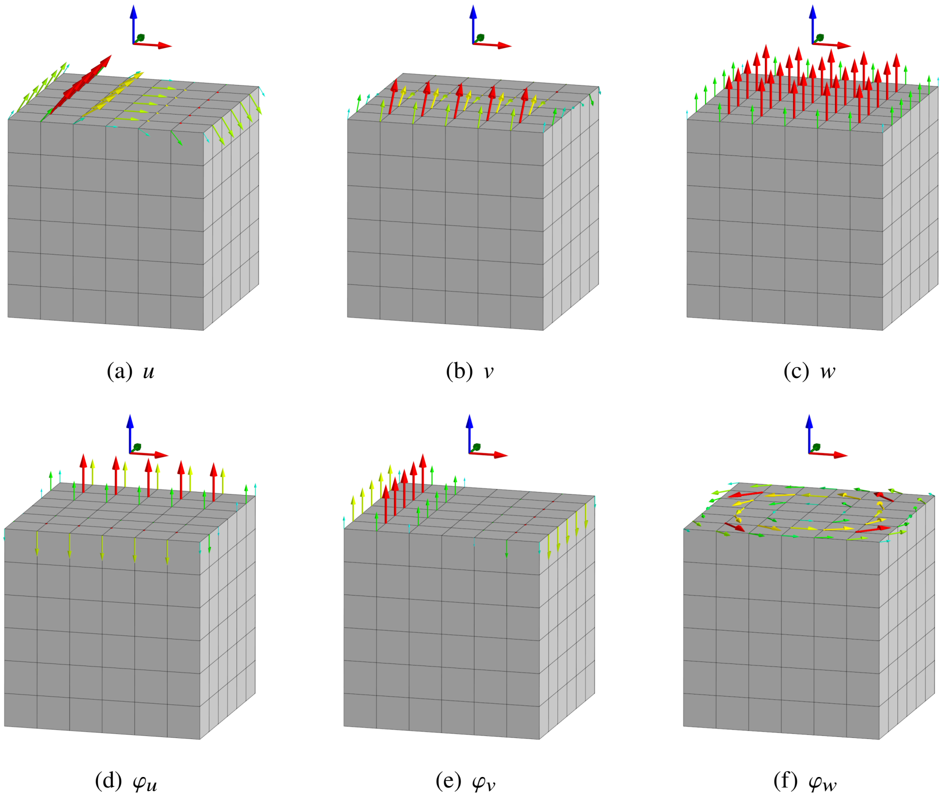
An interface on the top face of the cube with reference point in the centre of the face is shown in *Fig.* 4.8. The finite element mesh in this example is evenly spaced, what leads for the translational degrees of freedom to equal nodal DOF vectors for interior nodes of the face. The values for the nodes on the border of the face are half as high as the ones of the interior, because there are only two element faces contributing to each of those nodes. The nodes at the corners are connected to one element only and their values are therefore only one forth as high as the ones of the interior nodes. The vector fields of the elementary force distributions shown in *Fig.* 4.6 and *Fig.* 4.7 can be recognised in these nodal DOF vector distributions.



(a) u (b) v (c) w

(d) φ_u (e) φ_v (f) φ_w

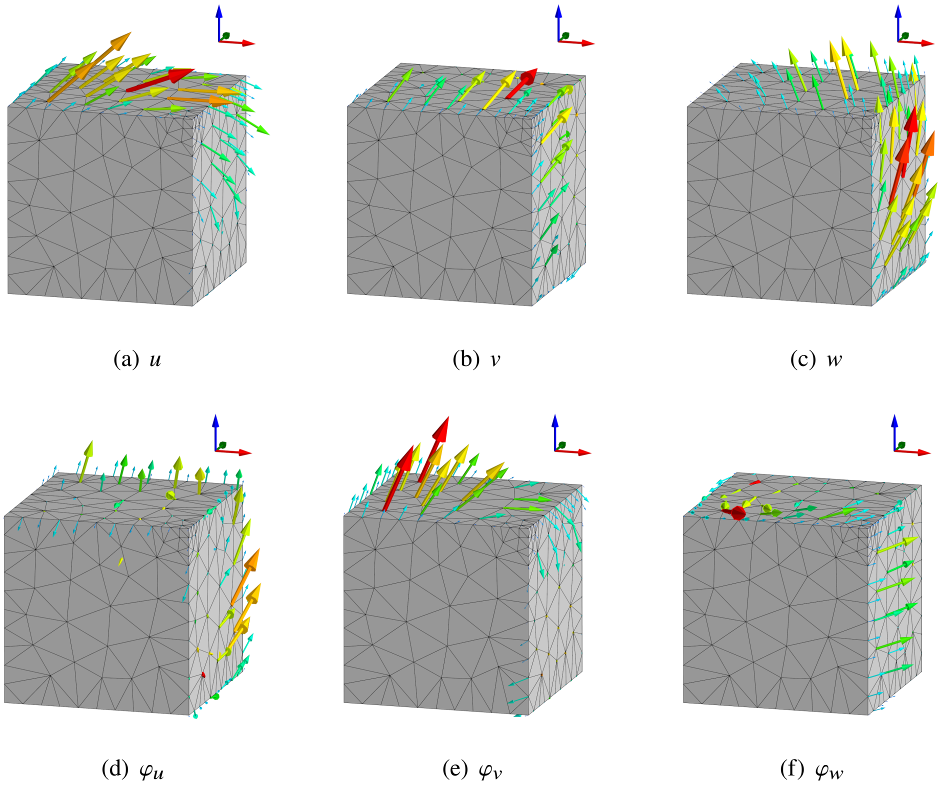
Figure 4.8: Nodal DOF vectors for all six degrees of freedom of a stationary interface on the top face of a cube with the reference point in the centre of area



(a) u (b) v (c) w

(d) φ_u (e) φ_v (f) φ_w

Figure 4.9: Nodal DOF vectors for all six degrees of freedom of a stationary interface on the top face of a cube with the reference point offset in w-direction


(a) u (b) v (c) w

(d) φ_u (e) φ_v (f) φ_w

Figure 4.10: Nodal DOF vectors for all six degrees of freedom of a stationary interface on the top and right face of an irregularly meshed cube with the reference point offset in u- and w-direction

In the second example, shown in *Fig.* 4.9, the interface's reference point is offset in w-direction, what leads to a superposition of the translational degrees of freedom u and v with the rotational degrees of freedom φ_v and φ_u, respectively. This corresponds exactly to the torques induced from a force acting on a cantilever from the face to the reference point.

The example from *Fig.* 4.10 shows an irregularly meshed cube, where tetrahedral elements are used. The interface's reference position is offset in both u and w direction, and two faces are used, the top and the right face. This example shows that the complexity of the force distribution grows rapidly with more complex geometries and reference point locations. However, the presented formalisms allow to find the correct distribution for the desired interface configuration.

4.3 Moving Interface Modelling by Trigonometric Interpolation

Moving axes of machine tools lead to changing positions of the coupling elements, and thus, the interfaces move depending on the axis positions and the finite element nodes affected by the interface change. Here, a method for the modelling of moving interfaces by trigonometric interpolation is presented.

4.3.1 Requirements on Moving Interface Models

The moving interfaces that occur in machine tool models usually change the position along a predefined path and the involved surfaces are known in advance. Examples for such interfaces are linear guides, ball screw spindles, or measurement scales.

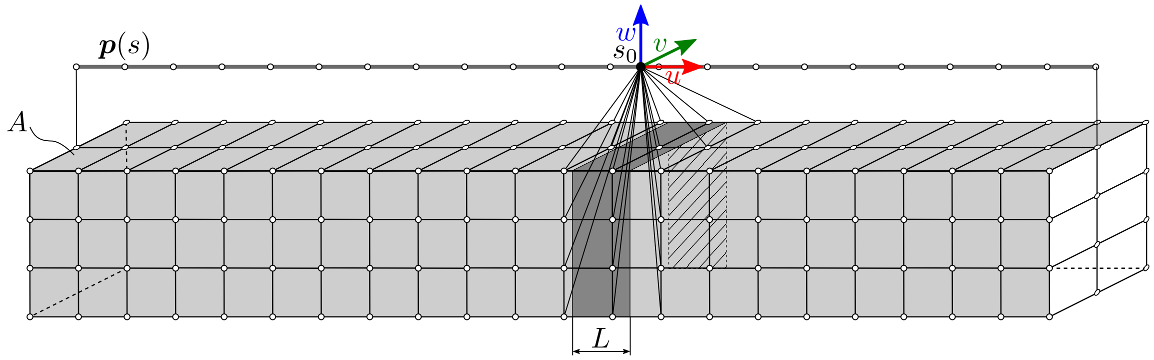


Figure 4.11: Example of a moving interface to three surfaces of a beam with rectangular cross section and figurative connection of a point on the path to the finite element nodes in the a restricted region

An illustration of such an interface on a beam with rectangular cross section is shown in *Fig.* 4.11. In this example, the front, top, and back faces denoted by A are part of the interface. The position of action changes along the path $\boldsymbol{p}$ which depends on the parameter $s \in [0, 1]$. For each parameter value s_0, restricting the resulting action on a segment of length L with the centre at s_0 has to be enabled. The resulting load has then to be distributed over the nodes of the element surfaces that intersect with the particular segment, as symbolised in *Fig.* 4.11. Because the geometry of the interfaced parts is usually simplified and the exact load transfer has not to be modelled, the particular distribution of the load in the segment is of secondary importance. The important requirements on the interface action are that

- the action is geometrically restricted to a width of approximately L,
- the resulting action corresponds to the desired one, and
- the resulting location of action corresponds to the desired one.

As with the stationary interfaces, all six spatial degrees of freedom have to be accessible through moving interfaces. Due to finite element discretisation of the structure, paths which are not straight can be modelled as chains of straight segments. Therefore, the derivation of the formalisms for straight paths suffices without loss of generality. In order to be compatible with model order reduction, it is required to reduce the number of inputs to the system to a minimum. The independent interaction with each finite element node, e.g., is not admissible, because each interface vector raises the order of the reduced system at least by one, because of the matching of the static behaviour.

4.3.2 Trigonometric Interpolation of a Weighting Function

The method presented here is based on trigonometric interpolation, i.e. Fourier series expansion, of a weighting function that depends on the path parameter s. The trigonometric interpolation $\hat{g}(s)$ with n_h harmonics for an arbitrary weighting function $g(s)$ is

$$\hat{g}(s) = \underbrace{a_0}_{g_0} + \sum_{k=1}^{n_h} \Big(a_k \underbrace{\cos(k\,2\pi\,s)}_{g_{ck}} + b_k \underbrace{\sin(k\,2\pi\,s)}_{g_{sk}} \Big) \tag{4.24}$$

with the Fourier coefficients that are calculated by

$$a_0 = \int_0^1 g(s)\,ds\,, \quad a_k = \int_0^1 g(s)\,\cos(k\,2\pi\,s)\,ds\,, \quad b_k = \int_0^1 g(s)\,\sin(k\,2\pi\,s)\,ds\,. \tag{4.25}$$

The weighting function used here in order to restrict the region of action is a trapezoidal function

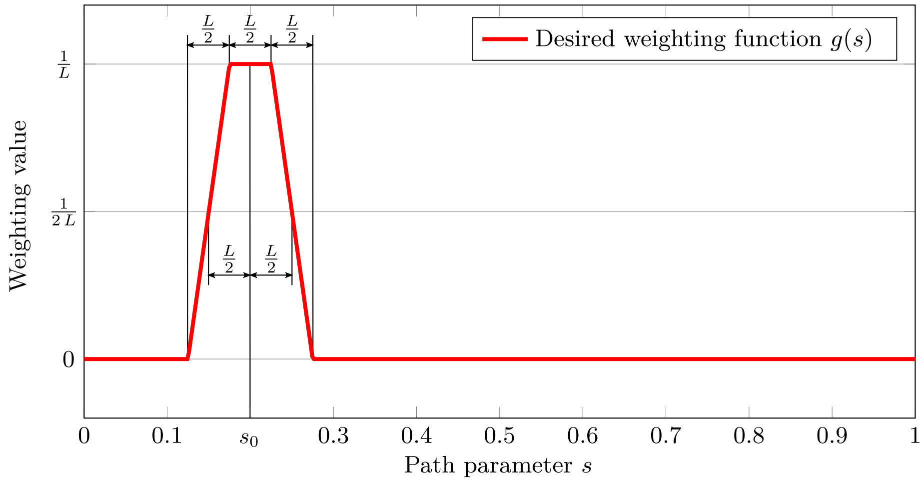


Figure 4.12: Trapezoidal weighting function from *Eq.* (4.26)

according to

$$g(s) = \begin{cases} \frac{1}{L} & \text{for } s_0 - \frac{1}{4}L \leq s \leq s_0 + \frac{1}{4}L \\ \frac{2}{L^2}\left(s - s_0 + \frac{3}{4}L\right) & \text{for } s_0 - \frac{3}{4}L \leq s < s_0 - \frac{1}{4}L \\ \frac{1}{L} - \frac{2}{L^2}\left(s - s_0 - \frac{1}{4}L\right) & \text{for } s_0 + \frac{1}{4}L < s \leq s_0 + \frac{3}{4}L \\ 0 & \text{else} \end{cases} \tag{4.26}$$

which is shown in *Fig.* 4.12. The choice of

$$L = \frac{1}{n_h} \tag{4.27}$$

leads to a good compromise between trigonometric interpolation quality and number of harmonics. The Fourier coefficients for this function, determined with *Eq.* (4.25), are

$$a_0 = 1 \tag{4.28}$$

for the constant term,

$$a_k = \frac{8}{L^2\, k^2\, \pi^2} \cos\left(\frac{k\,\pi\,L}{2}\right) \sin\left(\frac{k\,\pi\,L}{2}\right)^2 \cos(k\,2\pi\,s_0) \tag{4.29}$$

for the cosine terms, and

$$b_k = \frac{16}{L^2\, k^2\, \pi^2} \cos\left(\frac{k\,\pi\,L}{2}\right) \sin\left(\frac{k\,\pi\,L}{2}\right)^2 \cos(k\,\pi\,s_0)\, \sin(k\,\pi\,s_0) \tag{4.30}$$

for the sine terms. The resulting trigonometric interpolation and some weighted harmonics are shown in *Fig.* 4.13.

4.3.3 Elementary Force Distributions

The constant term g_0 and the harmonics g_{ck} and g_{sk} in *Eq.* (4.24) are invariant to the specific choice of a weighting function g. These weighting functions are used for the calculation of a set of elementary force distributions, i.e. nodal DOF vectors, which are used for the creation of the desired resulting

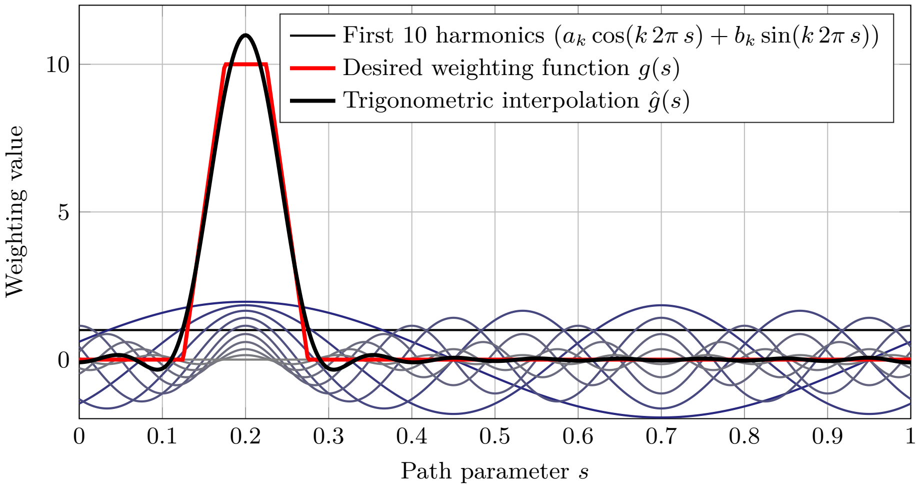


Figure 4.13: Trigonometric interpolation of the trapezoidal weighting function from *Eq.* (4.26)

Table 4.2: Elementary force distributions for moving interfaces

DOF	Description	Vector field $\boldsymbol{t}(u,v,w)$	Nodal DOF vectors
u	Translation in u-direction	$g_h\,\boldsymbol{e_u}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{h_u}} = \boldsymbol{e_u}\,\gamma^{[i]}_h$
v	Translation in v-direction	$g_h\,\boldsymbol{e_v}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{h_v}} = \boldsymbol{e_v}\,\gamma^{[i]}_h$
w	Translation in w-direction	$g_h\,\boldsymbol{e_w}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{h_w}} = \boldsymbol{e_w}\,\gamma^{[i]}_h$
φ_u	Rotation around u	$g_h\,(v\,\boldsymbol{e_w} - w\,\boldsymbol{e_v})$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{h_{\varphi u}}} = \boldsymbol{e_w}\,\gamma^{[i]}_{h_v} - \boldsymbol{e_v}\,\gamma^{[i]}_{h_w}$
φ_v	Rotation around v	$g_h\,w\,\boldsymbol{e_u}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{h_{\varphi v}}} = \boldsymbol{e_u}\,\gamma^{[i]}_{h_w}$
φ_w	Rotation around w	$-g_h\,v\,\boldsymbol{e_u}$	$\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{h_{\varphi w}}} = -\boldsymbol{e_u}\,\gamma^{[i]}_{h_v}$

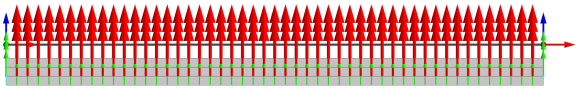
loads. Using the coefficients a_0, a_k, and b_k, the desired weighting function can then be varied during simulation. The procedure is similar to the one used for stationary interfaces in *Section* 4.2.2.

The elementary vector fields are of the form $g_h\big(s(u)\big) \cdot q(u,v,w) \cdot \boldsymbol{t_c}$, where g_h is the weighting function for the harmonic h of the trigonometric interpolation, either g_0, g_{ck}, or g_{sk}. They are built analogously to the ones for the stationary interfaces listed in *Tab.* 4.1. However, because the moving direction is the local coordinate u, the torque-generating force distributions with u-weighting have to be omitted. This leads to the elementary vector fields listed in *Tab.* 4.2.

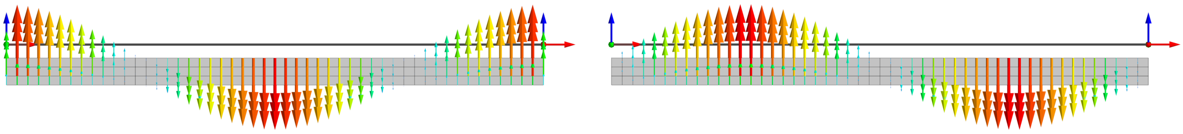
With the procedure presented in *Section* 4.1, the nodal weights $\gamma^{[i]}_h$, $\gamma^{[i]}_{h_u}$, $\gamma^{[i]}_{h_v}$, and $\gamma^{[i]}_{h_w}$ can be calculated. The nodal DOF vectors are then calculated according to the last column of *Tab.* 4.2. As an example, the nodal DOF vectors for $\boldsymbol{f}^{[i]}_{\boldsymbol{h_w}}$ are shown in *Fig.* 4.14 for the constant term and the first four harmonics.

4.3.4 Orthonormal Interfaces

The orthonormalisation of interfaces has been introduced in *Section* 4.2.3 for stationary interfaces. For moving interfaces, the same procedure is applied by means of the constant terms for all degrees of freedom. This leads to the resulting load matrix $\bar{\boldsymbol{F}}_{\boldsymbol{R}}$ and the orthonormalisation matrix $\boldsymbol{R_F}$ according to *Eq.* (4.21) and *Eq.* (4.22), respectively. The translational elements of the resulting load matrix are

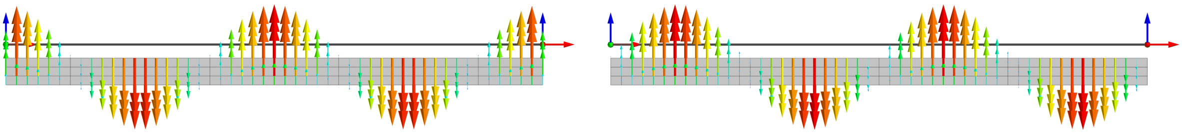


(a) Constant term $\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{0_w}}$



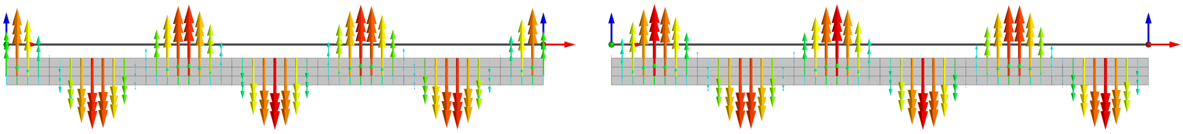
(b) 1. cosine term $\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{c1_w}}$

(c) 1. sine term $\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{s1_w}}$



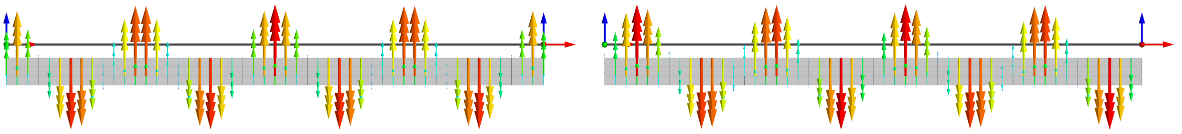
(d) 2. cosine term $\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{c2_w}}$

(e) 2. sine term $\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{s2_w}}$



(f) 3. cosine term $\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{c3_w}}$

(g) 3. sine term $\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{s3_w}}$



(h) 4. cosine term $\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{c4_w}}$

(i) 4. sine term $\bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{s4_w}}$

Figure 4.14: Nodal DOF vectors for the elementary force distribution in w-direction. Constant term and the first four harmonics.

calculated by

$$\bar{F}_{R_{a,b}} = \sum_{i=1}^{N_n} \boldsymbol{e_a} \bullet \bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{0_b}} \qquad a \in \{u, v, w\} \ , \quad b \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\} \tag{4.31}$$

and the rotational elements are calculated by

$$\bar{M}_{R_{a,b}} = \sum_{i=1}^{N_n} \boldsymbol{e_a} \bullet \left(\left(\boldsymbol{r}^{[i]} - \boldsymbol{p}(0.5) \right) \times \bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{0_b}} \right) \qquad a \in \{u, v, w\} \ , \quad b \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\} \ , \tag{4.32}$$

whereas the centre of the path $\boldsymbol{p}(0.5)$ is used as reference point.

The orthonormalisation matrix is then used for the normalisation of the nodal DOF vectors for all harmonics according to

$$\boldsymbol{f}^{[i]}_{\boldsymbol{h_a}} = \sum_{b} R_{F_{a,b}} \bar{\boldsymbol{f}}^{[i]}_{\boldsymbol{h_b}} \qquad a, b \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\} \ . \tag{4.33}$$

4.3.5 Nodal Values by Trigonometric Interpolation

The nodal DOF vectors from *Eq.* (4.33) corresponding to the harmonics of a Fourier series are used during simulation for the composition of the desired force distribution as a trigonometric interpolation according to

$$\boldsymbol{f}_{\boldsymbol{a}}^{[i]}(s_0) = \boldsymbol{f}_{\boldsymbol{0_a}}^{[i]} + \sum_{k=1}^{n_h}\left(a_k(s_0)\,\boldsymbol{f}_{\boldsymbol{ck_a}}^{[i]} + b_k(s_0)\,\boldsymbol{f}_{\boldsymbol{sk_a}}^{[i]}\right) \qquad a \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\} \ . \tag{4.34}$$

Fig. 4.15 shows the nodal values resulting from a trigonometric interpolation of a trapezoidal function together with the trapezoidal function and its trigonometric interpolation. Because of the evenly spaced finite element mesh in this example, there is a precise correspondence between the nodal values and the trigonometric interpolation of the weighting function.

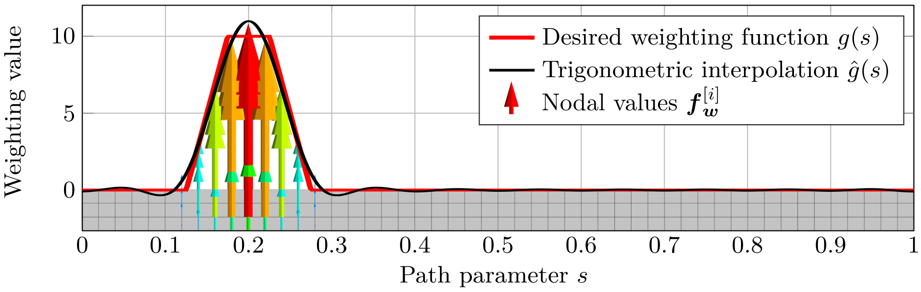


Figure 4.15: Nodal DOF vectors and weighting function for trigonometric interpolation of the trapezoidal function from *Eq.* (4.26)

In *Fig.* 4.16 the nodal values for all six spatial degrees of freedom of a moving interface are shown. Apparently, the resulting center of action lies on the path for the particular directions of action.

4.3.6 Verification of Resulting Action

In order to verify that the resulting action, i.e. the resulting force / torque application or displacement / rotation evaluation, of interfaces modelled using trigonometric interpolation are accurate and that the presented orthonormalisation is correct, the properties are evaluated for the bar model with three different meshes: a regular mesh which is hexahedral-based, an irregular mesh which is tetrahedral-based, and a coarse tetrahedral-based mesh, as shown in *Fig.* 4.17.

For the evaluation, the resulting load matrix is built for the moving interface's DOF vectors $\boldsymbol{f}_{\boldsymbol{a}}^{[i]}(s_0)$ for a multitude of positions s_0 along the path, with exception of the first and last 10%, which are affected by overflow of the weighting function. The resulting load matrix $\boldsymbol{F_R}(s_0)$ is then calculated analogously to *Eq.* (4.19) and *Eq.* (4.20). For a perfect interface, the resulting load matrix is the identity matrix. The deviation of the actual resulting load matrix to the identity matrix is a modelling error and is analysed in the following.

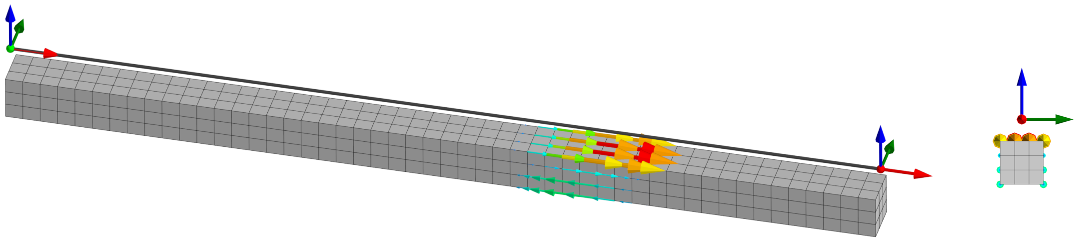
The maximum relative error of the magnitude of the action for each position is calculated by means of the values on the diagonal of the resulting load matrix according to

$$\hat{\varepsilon}_F(s_0) = \max\left|\mathrm{diag}\left(\boldsymbol{F_R}(s_0) - \boldsymbol{I}\right)\right| \tag{4.35}$$

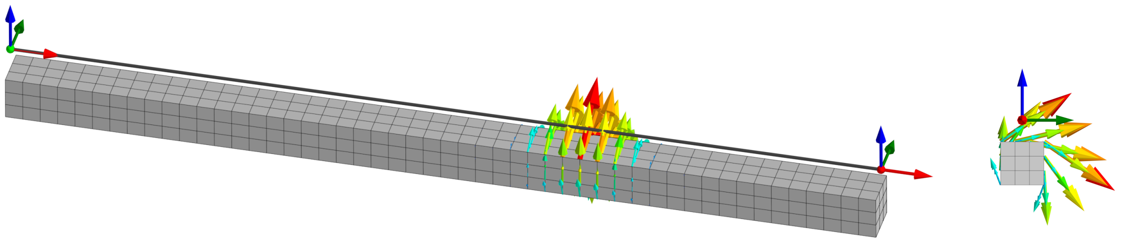
and is shown in *Fig.* 4.18(a). This value corresponds to the largest relative error of the resulting load for any of the six degrees of freedom of the interface. The results show that the load underlies very little fluctuation for the fine mesh variants. Even the very coarse mesh shows a relative error below 1% for the majority of the range.

In *Eq.* (4.35), only the direct error of the load for each degree of freedom is considered. However, cross-coupling of the interface's degrees of freedom is possible, i.e. the nodal values for a load in one direction introduce a load in another direction. These are the off-diagonal values of the resulting load matrix. Therefore, the maximum error of all elements of $\boldsymbol{F_R}(s_0)$ corresponds to the maximum orthonormality error according to

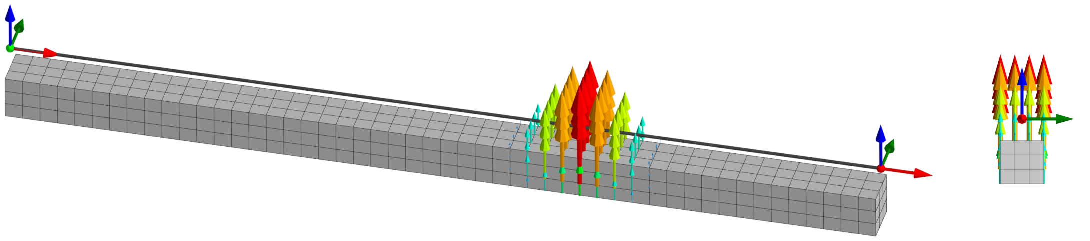
$$\hat{\varepsilon}_O(s_0) = \max\left|\boldsymbol{F_R}(s_0) - \boldsymbol{I}\right| \ . \tag{4.36}$$



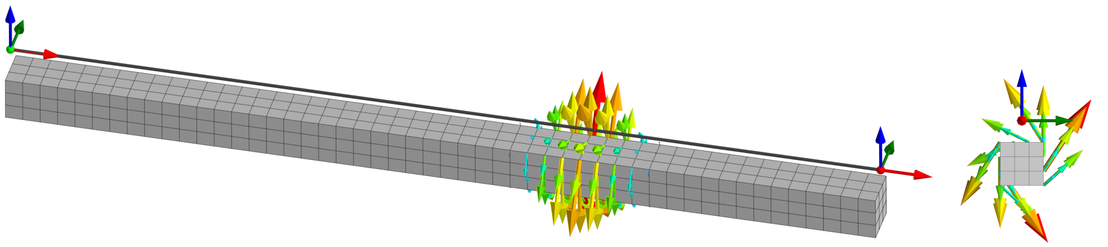
(a) u (perspective view) (b) u (side view)



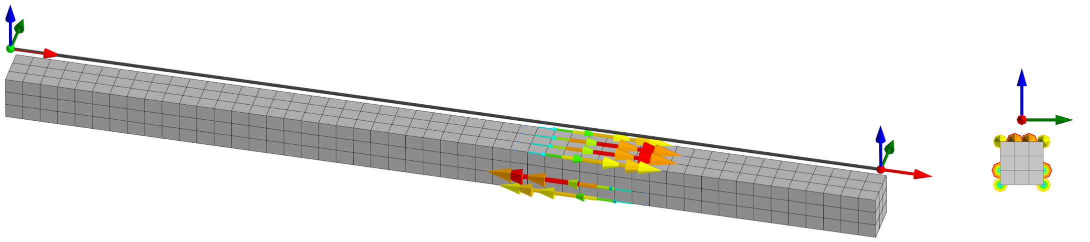
(c) v (perspective view) (d) v (side view)



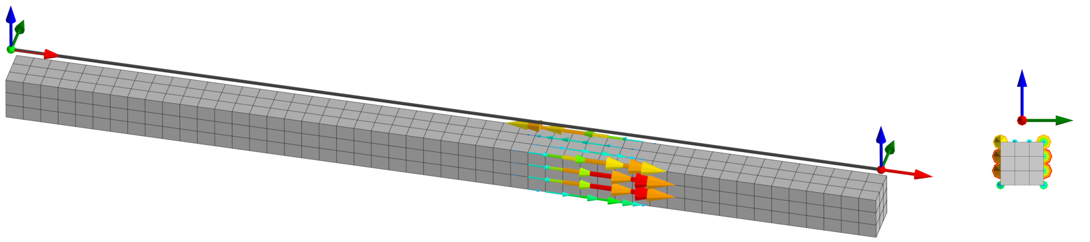
(e) w (perspective view) (f) w (side view)



(g) φ_u (perspective view) (h) φ_u (side view)

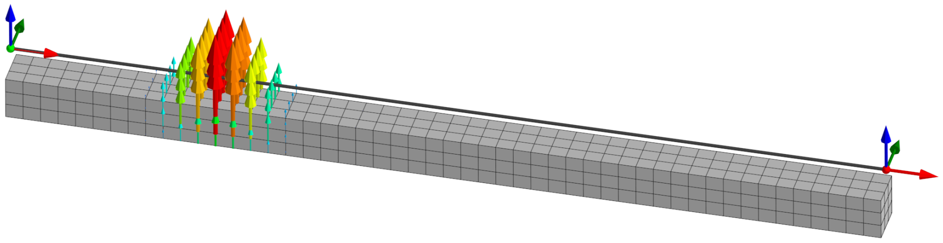


(i) φ_v (perspective view) (j) φ_v (side view)

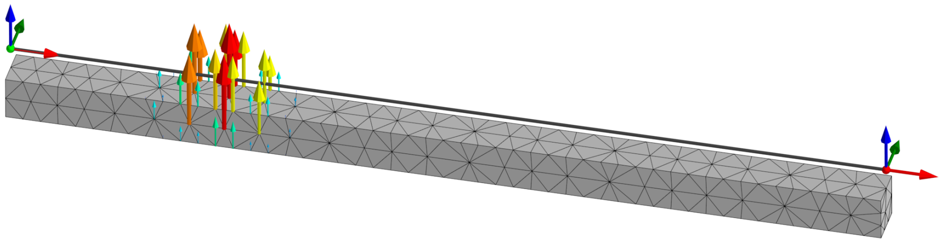


(k) φ_w (perspective view) (l) φ_w (side view)

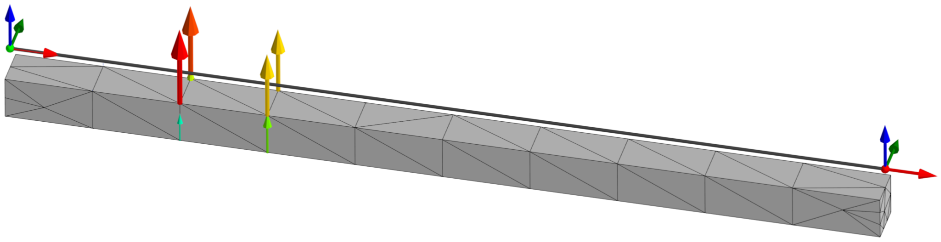
Figure 4.16: Nodal DOF vectors for all six degrees of freedom of a moving interface



(a) Hexahedral mesh



(b) Tetrahedral mesh



(c) Coarse tetrahedral mesh

Figure 4.17: Beam with three different meshes and the resulting nodal DOF vectors for a moving interface evaluated at the position $s_0 = 0.24$ for the degree of freedom w

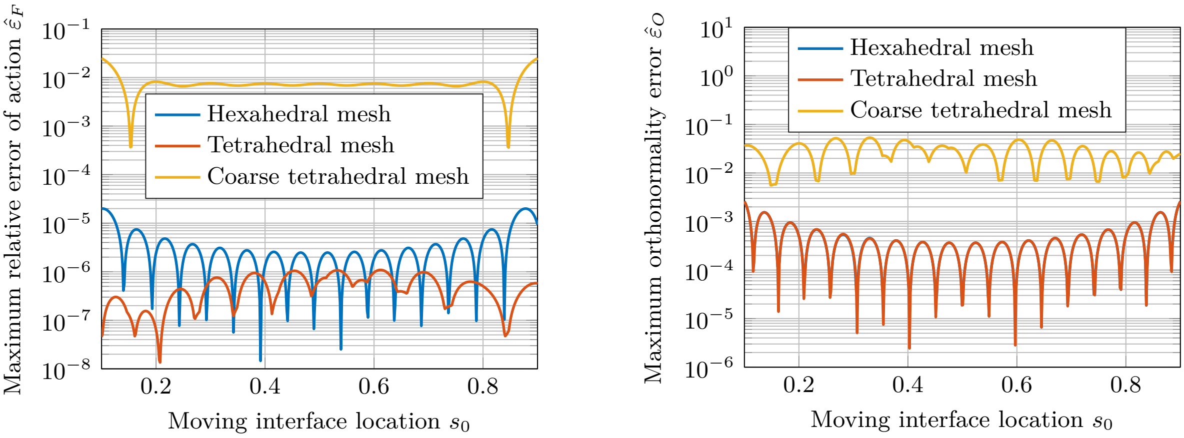
This is also a measure for the error of the location of the centre of action, because if the centre of action is shifted, torques are generated by the forces. The orthonormality errors for the three mesh variants are shown in *Fig.* 4.18(*b*). The maximum orthonormality error measures about 0.1% for the variants with fine meshes and the maximum value for the coarse mesh is ca. 5%. These are acceptable values compared to the quality of the mesh and the accuracy is appropriate for the application field of interest. Three main parameters influence the accuracy of the resulting loads: the mesh quality, the number of harmonics used, and the Gaussian quadrature order used for integration over the elements. For high-quality results, those three parameters can be enhanced.

4.4 Discussion of Interface Modelling

The methods presented enable modelling of interfaces with six degrees of freedom, both stationary and moving.

The stationary interfaces lead to an orthonormal resulting load matrix, and therefore, the location of the centre of action and the resulting load are exact. This kind of interfaces leads to six interface vectors, i.e. six inputs and outputs to the system, which have to be considered for model reduction, in order to assure a good matching quality.

Modelling of moving interfaces by means of trigonometric interpolation enables the geometric restriction of the action to a section of a predefined path. As weighting function for the geometrical restriction, a trapezoidal function is used. A good approximation is achieved if the number of harmonics



(a) Maximum relative error of action

(b) Maximum orthonormality error

Figure 4.18: Position dependent maximum errors for the moving interfaces from *Fig.* 4.17

n_h used corresponds to the reciprocal value of the relative width L of the load trapezoid according to *Eq.* (4.27). The resulting load matrix is not exactly orthogonal. However, it has been shown that the errors are reasonably small for appropriate finite element meshes.

The trigonometric interpolation is not restricted to trapezoidal functions. Every weighing function which can be approximated by trigonometric interpolation with the used number of harmonics can be modelled. Especially, the superposition of multiple functions by means of the same interface matrices is useful. This allows, e.g. to attach multiple linear guide carriages to one moving interface representing the rail.

The number of interface vectors to the finite element model is $(2 \cdot n_h + 1) \cdot 6$, i.e. six for each cosine, sine, and constant term. This leads to a decent total number of inputs that can be used in combination with the KMS-based model order reduction method presented in *Chapter* 3.

Arbitrary moving paths can be handled by segmentation of the path into straight segments. This is appropriate because also the geometry is discretised with the finite element mesh. Due to the trigonometric interpolation, the moving interface is periodic, i.e. the transition from the end to the start of the path is seamless. This is an advantage for moving interfaces acting around a circumference, as it is the case, e.g. for process forces in a turning process.

Because the vector fields used for the rotational degrees of freedom vary in the directions perpendicular to the path only but not in direction of the path, as listed in *Tab.* 4.2, the surface involved has to exhibit an extent in both directions perpendicular to the path. A plane surface, e.g. which lies in the u-v-plane, would not allow the introduction of a rotation around the v-direction, and thus, the resulting load matrix $\boldsymbol{F_R}$ would be singular and the interface could not be orthonormalised.

5 Composition of Flexible Multi-Body Systems

In this chapter, an application-oriented presentation of the use of model order reduction, stationary, and moving interfaces for the composition of a multi-body system is given.

Since linearised simulation is much more efficient in terms of simulation time and complexity, the thesis on hand concentrates on the linearised description of the system matrices. This suffices for the majority of simulation tasks and leads to a very efficient simulation work-flow. Because the dynamic behaviour of the structure of machine tools usually has much shorter time constants than the time used for considerable reorientation of a body, in the majority of cases, it is sufficient to analyse the dynamics of a machine tool linearised for a particular pose. This simplification does not affect the results of static, frequency response, and modal analyses. Transient simulations however, although possible for arbitrary starting positions, can be performed with linear approximation of rotations only. However, an efficient implementation of the non-linear equations of motion would be desirable for the analysis of, e.g. rotor dynamics or high-speed rotary motion with low-stiffness structural components like handling robots.

A flexible multi-body simulation framework is used as basis for the derivation of linearised system matrices of compositions of multiple bodies, generated for a specific configuration of the bodies in space. For this purpose, the system matrices of the bodies are set up for a desired pose, i.e. position and orientation.

The first step is to find system matrices for the re-oriented bodies by means of the original system matrices, the projection basis from model order reduction, and the orientation parameters. This topic is covered in *Section* 5.1. Subsequently, *Section* 5.2 deals with the assembly of the system matrices of multiple bodies to a composition, including assembled interface matrices and linear couplings, also called links. Last, in *Section* 5.3, link properties for some common coupling elements are derived.

5.1 Equations of Motion for a Flexible Body

Shabana [98] derived the kinematics and kinetics of flexible bodies by means of a floating frame of reference approach. The system of equations resulting from this approach has been presented in *Eq.* (2.7). This equation of motion is used as a basis and modified in the following. Because, for machine tool simulation, the deformation due to internal centrifugal and Coriolis forces is assumed to be of minor importance in comparison with the deformation due to external forces, the quadratic velocity vector $\boldsymbol{Q_v}$ from *Eq.* (2.7) is omitted in favour of linear equations of motion as in *Eq.* (3.1). The vector of external forces $\boldsymbol{Q}_{\boldsymbol{e}}^{i}$ is replaced by an interface matrix $\hat{\boldsymbol{B}}^{i}$ and an input force vector $\boldsymbol{u}^{i}$ according to

$$\boldsymbol{Q}_{\boldsymbol{e}}^{i} = \hat{\boldsymbol{B}}^{i} \boldsymbol{u}^{i} \; . \tag{5.1}$$

Furthermore, because every realisable connection in mechanical engineering has a finite stiffness, the algebraic constraints $\boldsymbol{C_q} \, \boldsymbol{\lambda}$ from *Eq.* (2.7) are discarded here.

In order to consider structural damping, additionally to *Eq.* (2.7), viscous structural damping is introduced through a damping matrix. The resulting equation of motion is

$$\underbrace{\begin{bmatrix}\hat{\boldsymbol{M}}^i_{tt} & \hat{\boldsymbol{M}}^i_{tr} & \hat{\boldsymbol{M}}^i_{tf}\\ & \hat{\boldsymbol{M}}^i_{rr} & \hat{\boldsymbol{M}}^i_{rf}\\ \text{symmetric} & & \hat{\boldsymbol{M}}^i_{ff}\end{bmatrix}}_{\hat{\boldsymbol{M}}^i}\underbrace{\begin{bmatrix}\ddot{\hat{\boldsymbol{x}}}^i_t\\ \ddot{\hat{\boldsymbol{x}}}^i_r\\ \ddot{\hat{\boldsymbol{x}}}^i_f\end{bmatrix}}_{\ddot{\hat{\boldsymbol{x}}}^i}+\underbrace{\begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \hat{\boldsymbol{D}}^i_{ff}\end{bmatrix}}_{\hat{\boldsymbol{D}}^i}\underbrace{\begin{bmatrix}\dot{\hat{\boldsymbol{x}}}^i_t\\ \dot{\hat{\boldsymbol{x}}}^i_r\\ \dot{\hat{\boldsymbol{x}}}^i_f\end{bmatrix}}_{\dot{\hat{\boldsymbol{x}}}^i}+\underbrace{\begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \hat{\boldsymbol{K}}^i_{ff}\end{bmatrix}}_{\hat{\boldsymbol{K}}^i}\underbrace{\begin{bmatrix}\hat{\boldsymbol{x}}^i_t\\ \hat{\boldsymbol{x}}^i_r\\ \hat{\boldsymbol{x}}^i_f\end{bmatrix}}_{\hat{\boldsymbol{x}}^i}=\underbrace{\begin{bmatrix}\hat{\boldsymbol{B}}^i_t\\ \hat{\boldsymbol{B}}^i_r\\ \hat{\boldsymbol{B}}^i_f\end{bmatrix}}_{\hat{\boldsymbol{B}}^i}\boldsymbol{u}^i \tag{5.2}$$

with the mass matrix $\hat{\boldsymbol{M}}^i$, the damping matrix $\hat{\boldsymbol{D}}^i$ the stiffness matrix $\hat{\boldsymbol{K}}^i$, and the interface matrix $\hat{\boldsymbol{B}}^i$, whereas both the system matrices and the state vector $\hat{\boldsymbol{x}}^i$ are split up into their components associated with translation, rotation, and deformation, denoted by the indices t, r, and f, respectively. The superscript i denotes the number of the body.

In the following, the compilation of all those components from model data available from the previous chapters is described.

5.1.1 Separation of Reference Motion and Deformation

Projection bases for model order reduction, in general, contain both rigid body movements as well as deformation behaviour. This holds also for a KMS basis $\boldsymbol{V}^i$ calculated as presented in *Chapter* 3. The floating frame of reference approach requires the separation of the body's flexible behaviour from its reference motion. This can be accomplished by subtracting the range of the rigid body modes from the reduction subspace spanned, e.g. by a KMS basis.

Rigid body modes for all three translations and three rotations in the Cartesian reference system of a body are to be specified. Therefore, the nodal DOF vectors for all nodes of a finite element mesh are defined explicitly. For the translations, the nodal DOF vectors are constant vectors corresponding to the unit vectors $\boldsymbol{e}^i_d$ of the finite element coordinate system of body i according to

$$\boldsymbol{x}^{[i_n]}_d = \boldsymbol{e}^i_d \qquad d \in \{x, y, z\}\ , \quad i_n = 1, 2, \ldots, N_n\ . \tag{5.3}$$

The nodal DOF vectors defining a rotational rigid body mode are tangential vectors at the nodes located at $\boldsymbol{r}^{[i_n]}$ around a point of rotation $\boldsymbol{r}^i_0$ and are calculated according to

$$\boldsymbol{x}^{[i_n]}_{\varphi d} = \boldsymbol{e}^i_d \times \left(\boldsymbol{r}^{[i_n]} - \boldsymbol{r}^i_0\right) \qquad d \in \{x, y, z\}\ , \quad i_n = 1, 2, \ldots, N_n \tag{5.4}$$

for each Cartesian axis $\boldsymbol{e}^i_d$. The rigid body modes are then built according to

$$\boldsymbol{\phi}_d = \sum_{i_n=1}^{N_n} \left[\boldsymbol{p}^{[i_n]}_x \quad \boldsymbol{p}^{[i_n]}_y \quad \boldsymbol{p}^{[i_n]}_z\right] \boldsymbol{x}^{[i_n]}_d \qquad d \in \left\{x, y, z, \varphi_x, \varphi_y, \varphi_z\right\} \tag{5.5}$$

with the allocation vectors $\boldsymbol{p}^{[i_n]}_d$ as presented in *Section* 4.1.4. With this definition, sets of translational and rotational rigid body modes $\boldsymbol{\Phi}^i_t$ and $\boldsymbol{\Phi}^i_r$, respectively, can be defined according to

$$\boldsymbol{\Phi}^i_t = \left[\boldsymbol{\phi}_x \quad \boldsymbol{\phi}_y \quad \boldsymbol{\phi}_z\right]\ , \quad \boldsymbol{\Phi}^i_r = \left[\boldsymbol{\phi}_{\varphi_x} \quad \boldsymbol{\phi}_{\varphi_y} \quad \boldsymbol{\phi}_{\varphi_z}\right]\ . \tag{5.6}$$

The chosen set of rigid body modes defined by the Cartesian reference system with the unit vectors $\boldsymbol{e}^i_d$ and the centre of rotation $\boldsymbol{r}^i_0$ defines the floating frame of reference for the body, i.e. a reference system that is bound to the body.

Projection of the original mass matrix $\boldsymbol{M}^i$ by means of the rigid body modes leads to matrices according to

$$\boldsymbol{\Phi}^{iT}_t \boldsymbol{M}^i \boldsymbol{\Phi}^i_t = \hat{\boldsymbol{M}}^i_{tt} = m^i \boldsymbol{I} \tag{5.7}$$

$$\boldsymbol{\Phi}^{iT}_r \boldsymbol{M}^i \boldsymbol{\Phi}^i_r = \hat{\boldsymbol{M}}^i_{rr} = \boldsymbol{J}^i \tag{5.8}$$

$$\boldsymbol{\Phi}^{iT}_t \boldsymbol{M}^i \boldsymbol{\Phi}^i_r = \hat{\boldsymbol{M}}^i_{tr} = m^i \tilde{\boldsymbol{r}}^i_S\ . \tag{5.9}$$

The relation between the block matrices and the inertia properties of the body can be found by comparison with the mass matrix derived by Lehner [63]. *Eq.* (5.7) corresponds to a diagonal matrix with the total mass m^i of the body on its diagonal, *Eq.* (5.8) corresponds to the inertia tensor $\boldsymbol{J}^i$ of the body with respect to the body reference system, and *Eq.* (5.9) corresponds to the inertia coupling between the translational and rotational degrees of freedom. Multiplication by the skew symmetric matrix $\tilde{\boldsymbol{r}}_{\boldsymbol{S}}^i$ from *Eq.* (5.9) corresponds to a cross product with the vector $\boldsymbol{r}_{\boldsymbol{S}}^i$, which is the position vector to the centre of mass of the body in the body reference system. Thus, this allows the determination of the centre of mass by means of the mass matrix and the rigid body modes.

Choosing the body reference frame to lie at the centre of mass is beneficial because it decouples the translational and rotational degrees of freedom. Therefore, if the rotation centre for *Eq.* (5.4) is chosen to lie on the centre of mass, $\boldsymbol{r}_{\boldsymbol{S}}^i$ is the zero vector and the rigid body mass matrix is

$$\begin{bmatrix}\boldsymbol{\Phi}_t^i & \boldsymbol{\Phi}_r^i\end{bmatrix}^T \boldsymbol{M}^i \begin{bmatrix}\boldsymbol{\Phi}_t^i & \boldsymbol{\Phi}_r^i\end{bmatrix} = \begin{bmatrix} m^i \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}^i \end{bmatrix} . \tag{5.10}$$

Now, the rigid body mode portions are to be eliminated from the projection basis $\boldsymbol{V}^i$ in order to separate reference motion and deformation. Therefore, *Algorithm* 7 from *Chapter* 3 can be called according to

$$\boldsymbol{V}_f^i = \textsc{ReduceRange}\left(\boldsymbol{V}^i, \begin{bmatrix}\boldsymbol{\Phi}_t^i & \boldsymbol{\Phi}_r^i\end{bmatrix}\right) . \tag{5.11}$$

Projection of the system matrices by the matrix $\boldsymbol{V}_f^i$ leads to the deformation-related system matrices according to

$$\hat{\boldsymbol{M}}_{ff}^i = \boldsymbol{V}_f^{i\,T} \boldsymbol{M}^i \boldsymbol{V}_f^i , \quad \hat{\boldsymbol{D}}_{ff}^i = \boldsymbol{V}_f^{i\,T} \boldsymbol{D}^i \boldsymbol{V}_f^i , \text{ and } \quad \hat{\boldsymbol{K}}_{ff}^i = \boldsymbol{V}_f^{i\,T} \boldsymbol{K}^i \boldsymbol{V}_f^i . \tag{5.12}$$

The coupling between the reference motion and deformation is defined by the matrices $\hat{\boldsymbol{M}}_{tf}^i$ and $\hat{\boldsymbol{M}}_{rf}^i$, which are the results of oblique projections by means of combinations of rigid body modes and the deformation-related model reduction basis according to

$$\hat{\boldsymbol{M}}_{tf}^i = \boldsymbol{\Phi}_t^{i\,T} \boldsymbol{M}^i \boldsymbol{V}_f^i , \text{ and } \quad \hat{\boldsymbol{M}}_{rf}^i = \boldsymbol{\Phi}_r^{i\,T} \boldsymbol{M}^i \boldsymbol{V}_f^i . \tag{5.13}$$

With this, all components of the system matrices from *Eq.* (5.2) are determined for the original orientation.

5.1.2 Orientation-Dependent System Matrices

It is desired to find the system matrices corresponding to a body oriented by a rotation matrix $\boldsymbol{A}^i$. The rotation matrix can be parametrised arbitrarily, e.g. using a rotation axis and angle, Euler angles, Euler parameters, Rodriguez parameters, or unit quaternions.

Because the set of rigid body modes defines the floating frame of reference, the system matrices can be re-oriented by means of rotation of the reference frame for the rigid body modes, what is accomplished with

$$\bar{\boldsymbol{\Phi}}_t^i = \boldsymbol{\Phi}_t^i \boldsymbol{A}^{i\,T} , \quad \bar{\boldsymbol{\Phi}}_r^i = \boldsymbol{\Phi}_r^i \boldsymbol{A}^{i\,T} . \tag{5.14}$$

Applying the transformation to the components of the mass matrix leads to

$$\hat{\boldsymbol{M}}^i_{\boldsymbol{tt}} = \bar{\boldsymbol{\Phi}}^{i\,T}_{\boldsymbol{t}} \boldsymbol{M}^i \bar{\boldsymbol{\Phi}}^i_{\boldsymbol{t}} = m^i \boldsymbol{I} \tag{5.15}$$

$$\hat{\boldsymbol{M}}^i_{\boldsymbol{rr}} = \bar{\boldsymbol{\Phi}}^{i\,T}_{\boldsymbol{r}} \boldsymbol{M}^i \bar{\boldsymbol{\Phi}}^i_{\boldsymbol{r}} = \boldsymbol{A}^i \boldsymbol{J}^i \boldsymbol{A}^{i\,T} \tag{5.16}$$

$$\hat{\boldsymbol{M}}^i_{\boldsymbol{tr}} = \bar{\boldsymbol{\Phi}}^{i\,T}_{\boldsymbol{t}} \boldsymbol{M}^i \bar{\boldsymbol{\Phi}}^i_{\boldsymbol{r}} = m^i_0 \boldsymbol{A}^i \tilde{\boldsymbol{r}}^i_{\boldsymbol{S}} \boldsymbol{A}^{i\,T} \tag{5.17}$$

$$\hat{\boldsymbol{M}}^i_{\boldsymbol{ff}} = \boldsymbol{V}^{i\,T}_{\boldsymbol{f}} \boldsymbol{M}^i \boldsymbol{V}^i_{\boldsymbol{f}} \tag{5.18}$$

$$\hat{\boldsymbol{M}}^i_{\boldsymbol{tf}} = \bar{\boldsymbol{\Phi}}^{i\,T}_{\boldsymbol{t}} \boldsymbol{M}^i \boldsymbol{V}^i_{\boldsymbol{f}} = \boldsymbol{A}^i \boldsymbol{\Phi}^{i\,T}_{\boldsymbol{t}} \boldsymbol{M}^i \boldsymbol{V}^i_{\boldsymbol{f}} \tag{5.19}$$

$$\hat{\boldsymbol{M}}^i_{\boldsymbol{rf}} = \bar{\boldsymbol{\Phi}}^{i\,T}_{\boldsymbol{r}} \boldsymbol{M}^i \boldsymbol{V}^i_{\boldsymbol{f}} = \boldsymbol{A}^i \boldsymbol{\Phi}^{i\,T}_{\boldsymbol{r}} \boldsymbol{M}^i \boldsymbol{V}^i_{\boldsymbol{f}} \,. \tag{5.20}$$

This shows that the inertia tensor $\boldsymbol{J}^i$ and the coupling between translation and rotation $\tilde{\boldsymbol{r}}^i_{\boldsymbol{S}}$ are rotated, that the couplings between the reference motion and the deformation $\hat{\boldsymbol{M}}^i_{\boldsymbol{tf}}$ and $\hat{\boldsymbol{M}}^i_{\boldsymbol{tr}}$ are changed, and that the components purely related to translation and deformation $\hat{\boldsymbol{M}}^i_{\boldsymbol{tt}}$ and $\hat{\boldsymbol{M}}^i_{\boldsymbol{ff}}$, respectively, are independent of the rotation.

Because the stiffness and the damping matrix are not affected by the rigid body modes, they do also not depend on the orientation of the body and are thus constant according to

$$\hat{\boldsymbol{K}}^i = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \hat{\boldsymbol{K}}^i_{\boldsymbol{ff}} \end{bmatrix} , \quad \hat{\boldsymbol{D}}^i = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \hat{\boldsymbol{D}}^i_{\boldsymbol{ff}} \end{bmatrix} \tag{5.21}$$

with $\hat{\boldsymbol{K}}^i_{\boldsymbol{ff}}$ and $\hat{\boldsymbol{D}}^i_{\boldsymbol{ff}}$ from *Eq.* (5.12).

5.1.3 Interfaces

An interface matrix $\boldsymbol{B}^i$ for the original finite element system is composed of n_i interfaces $\boldsymbol{B}^{i,k}$ with six degrees of freedom each according to

$$\boldsymbol{B}^i = \begin{bmatrix} \boldsymbol{B}^{i,1} & \boldsymbol{B}^{i,2} & \cdots & \boldsymbol{B}^{i,n_i} \end{bmatrix} , \tag{5.22}$$

where

$$\boldsymbol{B}^{i,k} = \begin{bmatrix} \boldsymbol{b}^{i,k}_{\boldsymbol{u}} & \boldsymbol{b}^{i,k}_{\boldsymbol{v}} & \boldsymbol{b}^{i,k}_{\boldsymbol{w}} & \boldsymbol{b}^{i,k}_{\boldsymbol{\varphi_u}} & \boldsymbol{b}^{i,k}_{\boldsymbol{\varphi_v}} & \boldsymbol{b}^{i,k}_{\boldsymbol{\varphi_w}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}^{i,k}_{\boldsymbol{t}} & \boldsymbol{B}^{i,k}_{\boldsymbol{r}} \end{bmatrix} \tag{5.23}$$

are orthonormal interfaces defined in an arbitrary local reference system (u, v, w) attached to the body and are created according to the procedure described in *Chapter* 4. The matrices $\boldsymbol{B}^{i,k}_{\boldsymbol{t}}$ and $\boldsymbol{B}^{i,k}_{\boldsymbol{r}}$ are the translational and rotational part of the interface, respectively.

5.1.3.1 Input Matrix

The input vector $\boldsymbol{u}^{i,k}$ belonging to interface k can be decomposed into its translational and rotational components, i.e. force and torque inputs in the interface coordinate system according to

$$\boldsymbol{u}^{i,k} = \begin{bmatrix} \boldsymbol{u}^{i,k}_{\boldsymbol{t}} \\ \boldsymbol{u}^{i,k}_{\boldsymbol{r}} \end{bmatrix} . \tag{5.24}$$

The interface's local reference system is defined by the unit vectors $\boldsymbol{e}^{i,k}_{\boldsymbol{u}}$, $\boldsymbol{e}^{i,k}_{\boldsymbol{v}}$, and $\boldsymbol{e}^{i,k}_{\boldsymbol{w}}$ and the origin $\boldsymbol{r}^{i,k}_{\boldsymbol{0}}$, both described in the body reference system. The matrix for the rotation from the interface reference system to the body reference system reads

$$\boldsymbol{A}^{i,k}_{\boldsymbol{0}} = \begin{bmatrix} \boldsymbol{e}^{i,k}_{\boldsymbol{u}} & \boldsymbol{e}^{i,k}_{\boldsymbol{v}} & \boldsymbol{e}^{i,k}_{\boldsymbol{w}} \end{bmatrix} . \tag{5.25}$$

The total force vector for the reduced system resulting from an input on interface k is

$$\hat{\boldsymbol{f}}^{i,k} = \begin{bmatrix} \hat{\boldsymbol{f}}_{\boldsymbol{t}}^{i,k} & \hat{\boldsymbol{f}}_{\boldsymbol{r}}^{i,k} & \hat{\boldsymbol{f}}_{\boldsymbol{f}}^{i,k} \end{bmatrix}^T = \hat{\boldsymbol{B}}^{i,k}\, \boldsymbol{u}^{i,k} \ . \tag{5.26}$$

The vectors $\hat{\boldsymbol{f}}_{\boldsymbol{t}}^{i,k}$ and $\hat{\boldsymbol{f}}_{\boldsymbol{r}}^{i,k}$ are the resulting force and torque vectors for the interface k, acting on the centre of mass of a body. These vectors are to be defined in the global reference system. The vector $\hat{\boldsymbol{f}}_{\boldsymbol{f}}^{i,k}$ consists of the deformation-related forces in the coordinates defined by the basis $\boldsymbol{V}_{\boldsymbol{f}}^{i}$.

The translational part of the force vector is built by consecutive rotation of the input force vector, first from interface coordinates to body coordinates, and then from body coordinates to global coordinates according to

$$\hat{\boldsymbol{f}}_{\boldsymbol{t}}^{i,k} = \boldsymbol{A}^i\, \boldsymbol{A}_{\boldsymbol{0}}^{i,k}\, \boldsymbol{u}_{\boldsymbol{t}}^{i,k} \ . \tag{5.27}$$

The rotational part is a sum of the rotated torque input vector and the torque generated by the input force vector with an offset from the centre of mass to the centre of action of the interface according to

$$\hat{\boldsymbol{f}}_{\boldsymbol{r}}^{i,k} = \underbrace{\boldsymbol{A}^i\, \boldsymbol{A}_{\boldsymbol{0}}^{i,k}\, \boldsymbol{u}_{\boldsymbol{r}}^{i,k}}_{\text{torque input}} + \underbrace{\bar{\boldsymbol{r}}_{\boldsymbol{0}}^{i,k} \times \left(\boldsymbol{A}^i\, \boldsymbol{A}_{\boldsymbol{0}}^{i,k}\, \boldsymbol{u}_{\boldsymbol{t}}^{i,k}\right)}_{\text{force cantilever}} \ . \tag{5.28}$$

Herein, $\bar{\boldsymbol{r}}_{\boldsymbol{0}}^{i,k}$ is the position vector $\boldsymbol{r}_{\boldsymbol{0}}^{i,k}$, rotated into the global reference system according to

$$\bar{\boldsymbol{r}}_{\boldsymbol{0}}^{i,k} = \boldsymbol{A}^i\, \boldsymbol{r}_{\boldsymbol{0}}^{i,k} \ . \tag{5.29}$$

The cross product can be expressed as matrix multiplication using the skew symmetric matrix $\tilde{\bar{\boldsymbol{r}}}_{\boldsymbol{0}}^{i,k}$ built with the elements of $\bar{\boldsymbol{r}}_{\boldsymbol{0}}^{i,k}$, leading to

$$\hat{\boldsymbol{f}}_{\boldsymbol{r}}^{i,k} = \boldsymbol{A}^i\, \boldsymbol{A}_{\boldsymbol{0}}^{i,k}\, \boldsymbol{u}_{\boldsymbol{r}}^{i,k} + \tilde{\bar{\boldsymbol{r}}}_{\boldsymbol{0}}^{i,k}\, \boldsymbol{A}^i\, \boldsymbol{A}_{\boldsymbol{0}}^{i,k}\, \boldsymbol{u}_{\boldsymbol{t}}^{i,k} \ . \tag{5.30}$$

Last, the deformation-related forces can be expressed by the interface matrix projected by the reduction basis with eliminated rigid body behaviour according to

$$\hat{\boldsymbol{f}}_{\boldsymbol{f}}^{i,k} = {\boldsymbol{V}_{\boldsymbol{f}}^{i}}^T\, \boldsymbol{B}^{i,k}\, \boldsymbol{u}^{i,k} = {\boldsymbol{V}_{\boldsymbol{f}}^{i}}^T\, \boldsymbol{B}_{\boldsymbol{t}}^{i,k}\, \boldsymbol{u}_{\boldsymbol{t}}^{i,k} + {\boldsymbol{V}_{\boldsymbol{f}}^{i}}^T\, \boldsymbol{B}_{\boldsymbol{r}}^{i,k}\, \boldsymbol{u}_{\boldsymbol{r}}^{i,k} \ . \tag{5.31}$$

Expressing *Eq.* (5.27), *Eq.* (5.30), and *Eq.* (5.31) as one matrix multiplication leads to the desired interface matrix for the reduced system

$$\hat{\boldsymbol{B}}^{i,k} = \begin{bmatrix} \hat{\boldsymbol{B}}_{\boldsymbol{t}}^{i,k} \\ \hat{\boldsymbol{B}}_{\boldsymbol{r}}^{i,k} \\ \hat{\boldsymbol{B}}_{\boldsymbol{f}}^{i,k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}^i\, {\boldsymbol{A}_{\boldsymbol{0}}}^{i,k} & \boldsymbol{0} \\ \tilde{\bar{\boldsymbol{r}}}_{\boldsymbol{0}}^{i,k}\, \boldsymbol{A}^i\, {\boldsymbol{A}_{\boldsymbol{0}}}^{i,k} & \boldsymbol{A}^i\, {\boldsymbol{A}_{\boldsymbol{0}}}^{i,k} \\ {\boldsymbol{V}_{\boldsymbol{f}}^{i}}^T\, \boldsymbol{B}_{\boldsymbol{t}}^{i,k} & {\boldsymbol{V}_{\boldsymbol{f}}^{i}}^T\, \boldsymbol{B}_{\boldsymbol{r}}^{i,k} \end{bmatrix} \ . \tag{5.32}$$

5.1.3.2 Output Matrix

In *Chapter* 4 it was shown that an input matrix $\boldsymbol{B}^i$ can be used equivalently as an output matrix of a finite element model as $\boldsymbol{C}^i = {\boldsymbol{B}^i}^T$. Following, it is shown that this also holds for the interfaces of reduced models in an arbitrary orientation.

The translational output of a system's interface is composed of the translation of the body, the cantilever part, resulting from rotating the position vector of the interface with the body, and the deformation of the body according to

$$\hat{\boldsymbol{y}}_{\boldsymbol{t}} = \underbrace{{\boldsymbol{A}_{\boldsymbol{0}}^{i,k}}^T {\boldsymbol{A}^i}^T\, \hat{\boldsymbol{x}}_{\boldsymbol{t}}^i}_{\text{body translation}} + \underbrace{{\boldsymbol{A}_{\boldsymbol{0}}^{i,k}}^T {\boldsymbol{A}^i}^T \left(\hat{\boldsymbol{x}}_{\boldsymbol{r}}^i \times \bar{\boldsymbol{r}}_{\boldsymbol{0}}^{i,k}\right)}_{\text{cantilever part}} + \underbrace{\left({\boldsymbol{V}_{\boldsymbol{f}}^{i}}^T\, \boldsymbol{B}_{\boldsymbol{t}}^{i,k}\right)^T \hat{\boldsymbol{x}}_{\boldsymbol{f}}^i}_{\text{structure deformation}} \ . \tag{5.33}$$

In accordance with the linearised system matrices, the cantilever part linearises the rotation as a tangential vector. Both the translation and cantilever part are transformed from global coordinates to body coordinates and further to local interface coordinates by the transformation matrix $\boldsymbol{A}_{\mathbf{0}}^{i,k^T}\boldsymbol{A}^{i^T}$.

The cantilever part can be reformulated as

$$\hat{\boldsymbol{x}}_{\boldsymbol{r}}^{i} \times \bar{\boldsymbol{r}}_{\mathbf{0}}^{i,k} = -\bar{\boldsymbol{r}}_{\mathbf{0}}^{i,k} \times \hat{\boldsymbol{x}}_{\boldsymbol{r}}^{i} = -\tilde{\bar{\boldsymbol{r}}}_{\mathbf{0}}^{i,k}\,\hat{\boldsymbol{x}}_{\boldsymbol{r}}^{i} = \tilde{\bar{\boldsymbol{r}}}_{\mathbf{0}}^{i,k^T}\,\hat{\boldsymbol{x}}_{\boldsymbol{r}}^{i}\ . \tag{5.34}$$

By reversing the order of the cross product in the cantilever part, the sign changes. Using the skew symmetric matrix $\tilde{\bar{\boldsymbol{r}}}_{\mathbf{0}}^{i,k}$, the cross product can again be expressed as matrix multiplication. Last, by transposing the skew symmetric matrix, the sign changes again. With this, *Eq.* (5.33) can be rewritten as

$$\hat{\boldsymbol{y}}_{\boldsymbol{t}} = \boldsymbol{A}_{\mathbf{0}}^{i,k^T}\boldsymbol{A}^{i^T}\,\hat{\boldsymbol{x}}_{\boldsymbol{t}}^{i} + \boldsymbol{A}_{\mathbf{0}}^{i,k^T}\boldsymbol{A}^{i^T}\,\tilde{\bar{\boldsymbol{r}}}_{\mathbf{0}}^{i,k^T}\,\hat{\boldsymbol{x}}_{\boldsymbol{r}}^{i} + \left(\boldsymbol{V}_{\boldsymbol{f}}^{i\,T}\,\boldsymbol{B}_{\boldsymbol{t}}^{i,k}\right)^{T}\hat{\boldsymbol{x}}_{\boldsymbol{f}}^{i}\ . \tag{5.35}$$

The rotation of an interface is given by the rotation of a body and the rotation of the interface due to deformation according to

$$\hat{\boldsymbol{y}}_{\boldsymbol{r}} = \underbrace{\boldsymbol{A}_{\mathbf{0}}^{i,k^T}\boldsymbol{A}^{i^T}\,\hat{\boldsymbol{x}}_{\boldsymbol{r}}^{i}}_{\text{body rotation}} + \underbrace{\left(\boldsymbol{V}_{\boldsymbol{f}}^{i\,T}\,\boldsymbol{B}_{\boldsymbol{r}}^{i,k}\right)^{T}\hat{\boldsymbol{x}}_{\boldsymbol{f}}^{i}}_{\text{structure deformation}}\ . \tag{5.36}$$

Using *Eq.* (5.35) and *Eq.* (5.36), the output vector can be written as the matrix multiplication

$$\hat{\boldsymbol{y}}^{i,k} = \begin{bmatrix} \boldsymbol{A}_{\mathbf{0}}^{i,k^T}\boldsymbol{A}^{i^T} & \boldsymbol{A}_{\mathbf{0}}^{i,k^T}\boldsymbol{A}^{i^T}\,\tilde{\bar{\boldsymbol{r}}}_{\mathbf{0}}^{i,k^T} & \left(\boldsymbol{V}_{\boldsymbol{f}}^{i\,T}\,\boldsymbol{B}_{\boldsymbol{t}}^{i,k}\right)^{T} \\ \mathbf{0} & \boldsymbol{A}_{\mathbf{0}}^{i,k^T}\boldsymbol{A}^{i^T} & \left(\boldsymbol{V}_{\boldsymbol{f}}^{i\,T}\,\boldsymbol{B}_{\boldsymbol{r}}^{i,k}\right)^{T} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_{\boldsymbol{t}}^{i} \\ \hat{\boldsymbol{x}}_{\boldsymbol{r}}^{i} \\ \hat{\boldsymbol{x}}_{\boldsymbol{f}}^{i} \end{bmatrix} \tag{5.37}$$

and the output matrix turns out to be equal to the transpose of the input matrix from *Eq.* (5.32), and thus,

$$\hat{\boldsymbol{y}}^{i,k} = \hat{\boldsymbol{B}}^{i,k^T}\,\hat{\boldsymbol{x}}^{i} \tag{5.38}$$

holds.

5.2 Assembly of Flexible Multi-Body Systems

So far, the equations of motion and the system matrices of single bodies of a multi-body system have been derived. Subsequently, it will be shown how these matrices can be combined to a composition of multiple bodies.

5.2.1 System Matrices of a Composition of Multiple Bodies

In order to define the state vector $\hat{\boldsymbol{x}}$ for a flexible multi-body system, the state vectors of all n_b bodies are concatenated as

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \hat{\boldsymbol{x}}^{1} \\ \hat{\boldsymbol{x}}^{2} \\ \vdots \\ \hat{\boldsymbol{x}}^{n_b} \end{bmatrix}\ . \tag{5.39}$$

There is no coupling between the mass matrices of different bodies, and therefore, the composed mass matrix has the single mass matrices on its diagonal and is zero otherwise, according to

$$\hat{\boldsymbol{M}} = \begin{bmatrix} \hat{\boldsymbol{M}}^1 & & & \\ & \hat{\boldsymbol{M}}^2 & & \\ & & \ddots & \\ & & & \hat{\boldsymbol{M}}^{n_b} \end{bmatrix} . \tag{5.40}$$

In general, the damping and stiffness matrices, however, have a coupling between the individual bodies' matrices, because couplings like, e.g. bearings, linear guides, ball screws, usually act on two bodies. The couplings which are also denoted as links, provided they are linear, can be expressed by means of stiffness matrices $\hat{\boldsymbol{K}}^l$ and damping matrices $\hat{\boldsymbol{D}}^l$, whose derivation will be discussed in the next section. For a set of n_l links, the damping matrix and stiffness matrix, respectively, result to

$$\hat{\boldsymbol{D}} = \begin{bmatrix} \hat{\boldsymbol{D}}^1 & & & \\ & \hat{\boldsymbol{D}}^2 & & \\ & & \ddots & \\ & & & \hat{\boldsymbol{D}}^{n_b} \end{bmatrix} + \sum_{l=1}^{n_l} \hat{\boldsymbol{D}}^l , \qquad \hat{\boldsymbol{K}} = \begin{bmatrix} \hat{\boldsymbol{K}}^1 & & & \\ & \hat{\boldsymbol{K}}^2 & & \\ & & \ddots & \\ & & & \hat{\boldsymbol{K}}^{n_b} \end{bmatrix} + \sum_{l=1}^{n_l} \hat{\boldsymbol{K}}^l . \tag{5.41}$$

An interface matrix $\hat{\boldsymbol{B}}^{i,k}$ of a body can be extended to match the composed state vector according to

$$\hat{\boldsymbol{B}}_{\boldsymbol{G}}^{i,k} = \begin{bmatrix} \boldsymbol{0} & \cdots & \boldsymbol{0} & \hat{\boldsymbol{B}}^{i,k^T} & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix}^T . \tag{5.42}$$

5.2.2 Links

A link is defined by a source interface $\hat{\boldsymbol{B}}_{\boldsymbol{S}}^l$ and a target interface $\hat{\boldsymbol{B}}_{\boldsymbol{T}}^l$, which can be any out of all defined interfaces, defined as in *Eq.* (5.42). The interaction between displacements or velocities with the forces of those two interfaces is characterised by the stiffness and damping matrices $\hat{\boldsymbol{K}}^l$ and $\hat{\boldsymbol{D}}^l$, respectively. The displacements of the target and source interfaces are

$$\hat{\boldsymbol{y}}_{\boldsymbol{T}}^l = \hat{\boldsymbol{B}}_{\boldsymbol{T}}^{l\,T}\, \hat{\boldsymbol{x}} , \quad \hat{\boldsymbol{y}}_{\boldsymbol{S}}^l = \hat{\boldsymbol{B}}_{\boldsymbol{S}}^{l\,T}\, \hat{\boldsymbol{x}} , \qquad \hat{\boldsymbol{y}}_{\boldsymbol{T}}^l, \hat{\boldsymbol{y}}_{\boldsymbol{S}}^l \in \mathbb{R}^6 \tag{5.43}$$

respectively. By introducing stiffness matrices $\boldsymbol{K}_{\boldsymbol{TT}}^l$ and $\boldsymbol{K}_{\boldsymbol{TS}}^l$, the reaction force on the target interface can be calculated by

$$\hat{\boldsymbol{f}}_{\boldsymbol{T}}^l = \boldsymbol{K}_{\boldsymbol{TT}}^l\, \hat{\boldsymbol{y}}_{\boldsymbol{T}}^l + \boldsymbol{K}_{\boldsymbol{TS}}^l\, \hat{\boldsymbol{y}}_{\boldsymbol{S}}^l \qquad \boldsymbol{K}_{\boldsymbol{TT}}^l, \boldsymbol{K}_{\boldsymbol{TS}}^l \in \mathbb{R}^{6\times 6} , \quad \hat{\boldsymbol{f}}_{\boldsymbol{T}}^l \in \mathbb{R}^6 \tag{5.44}$$

and, likewise, the reaction on the source interface is

$$\hat{\boldsymbol{f}}_{\boldsymbol{S}}^l = \boldsymbol{K}_{\boldsymbol{ST}}^l\, \hat{\boldsymbol{y}}_{\boldsymbol{S}}^l + \boldsymbol{K}_{\boldsymbol{TT}}^l\, \hat{\boldsymbol{y}}_{\boldsymbol{T}}^l . \qquad \boldsymbol{K}_{\boldsymbol{ST}}^l, \boldsymbol{K}_{\boldsymbol{SS}}^l \in \mathbb{R}^{6\times 6} , \quad \hat{\boldsymbol{f}}_{\boldsymbol{S}}^l \in \mathbb{R}^6 \tag{5.45}$$

Transforming these forces to the state coordinates leads to the state force vector $\hat{\boldsymbol{f}}^l$ according to

$$\hat{\boldsymbol{f}}^l = \hat{\boldsymbol{B}}_{\boldsymbol{T}}^l\, \hat{\boldsymbol{f}}_{\boldsymbol{T}}^l + \hat{\boldsymbol{B}}_{\boldsymbol{S}}^l\, \hat{\boldsymbol{f}}_{\boldsymbol{S}}^l . \tag{5.46}$$

Putting all this together, the elastic link can be described by the matrix equation

$$\hat{\boldsymbol{f}}^l = \begin{bmatrix} \hat{\boldsymbol{B}}_{\boldsymbol{T}}^l & \hat{\boldsymbol{B}}_{\boldsymbol{S}}^l \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{TT}}^l & \boldsymbol{K}_{\boldsymbol{TS}}^l \\ \boldsymbol{K}_{\boldsymbol{ST}}^l & \boldsymbol{K}_{\boldsymbol{SS}}^l \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{B}}_{\boldsymbol{T}}^l & \hat{\boldsymbol{B}}_{\boldsymbol{S}}^l \end{bmatrix}^T \hat{\boldsymbol{x}} , \tag{5.47}$$

from which the link stiffness matrix can be determined as

$$\hat{\boldsymbol{K}}^l = \begin{bmatrix} \hat{\boldsymbol{B}}_{\boldsymbol{T}}^l & \hat{\boldsymbol{B}}_{\boldsymbol{S}}^l \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{TT}}^l & \boldsymbol{K}_{\boldsymbol{TS}}^l \\ \boldsymbol{K}_{\boldsymbol{ST}}^l & \boldsymbol{K}_{\boldsymbol{SS}}^l \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{B}}_{\boldsymbol{T}}^l & \hat{\boldsymbol{B}}_{\boldsymbol{S}}^l \end{bmatrix}^T . \tag{5.48}$$

In the case that one wants to couple an interface elastically to the ground, i.e. the inertial system, components of the stiffness matrix related to the source interface are omitted and the stiffness matrix results to

$$\hat{\boldsymbol{K}}^l = \hat{\boldsymbol{B}}_{\boldsymbol{T}}^l \, \boldsymbol{K}_{\boldsymbol{TT}}^l \, \hat{\boldsymbol{B}}_{\boldsymbol{T}}^{l\,T} . \tag{5.49}$$

By replacing $\hat{\boldsymbol{x}}$ by $\dot{\hat{\boldsymbol{x}}}$ in *Eq.* (5.43), the link damping matrix can be determined as

$$\hat{\boldsymbol{D}}^l = \begin{bmatrix} \hat{\boldsymbol{B}}_{\boldsymbol{T}}^l & \hat{\boldsymbol{B}}_{\boldsymbol{S}}^l \end{bmatrix} \begin{bmatrix} \boldsymbol{D}_{\boldsymbol{TT}}^l & \boldsymbol{D}_{\boldsymbol{TS}}^l \\ \boldsymbol{D}_{\boldsymbol{ST}}^l & \boldsymbol{D}_{\boldsymbol{SS}}^l \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{B}}_{\boldsymbol{T}}^l & \hat{\boldsymbol{B}}_{\boldsymbol{S}}^l \end{bmatrix}^T . \tag{5.50}$$

Furthermore, a link can be defined as input or output, e.g. a motor, encoder, or scale, by building the difference between the target and the source interface matrices according to

$$\hat{\boldsymbol{B}}^l = \hat{\boldsymbol{B}}_{\boldsymbol{T}}^l - \hat{\boldsymbol{B}}_{\boldsymbol{S}}^l . \tag{5.51}$$

A load on a link, then, results to an actio load to the target interface and a reactio load on the source interface, and an output gives the relative displacement or velocity between the two interfaces.

5.3 Link Properties for Special Machine Parts

For an efficient modelling of machine tools, it is necessary to provide a catalogue of link properties, consisting of stiffness and damping matrices for different standard coupling elements.

Therefore, equations describing the relation between the displacement and the load of the source and target interfaces are to be derived. If these equations can be written in matrix-form, then the stiffness components of the link can be found by comparison with

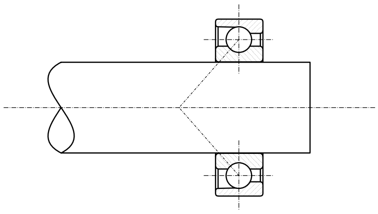
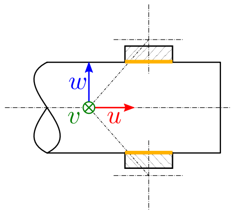
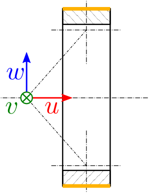
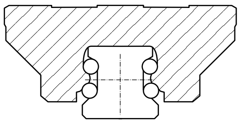
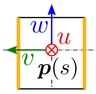
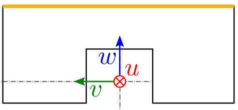
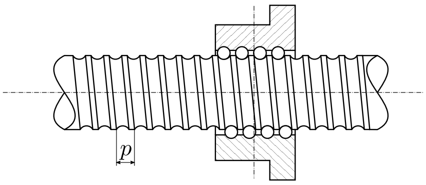
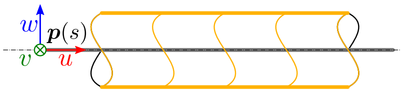
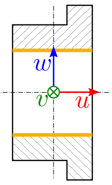
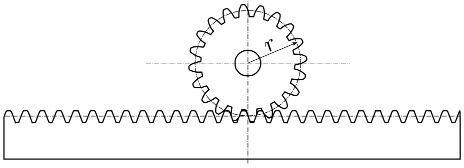
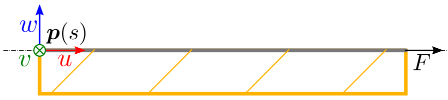
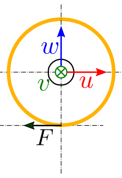
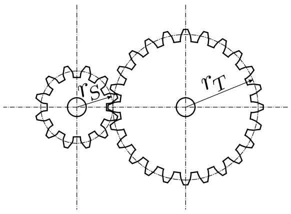
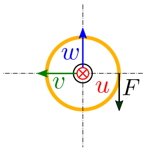
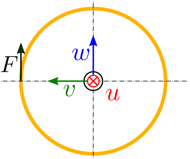
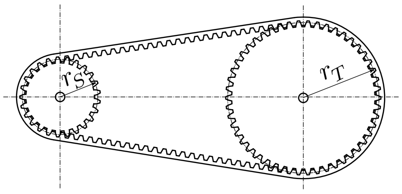
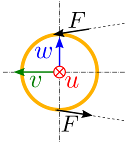
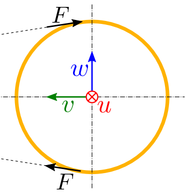
$$\begin{bmatrix} \hat{\boldsymbol{f}}_{\boldsymbol{T}}^l \\ \hat{\boldsymbol{f}}_{\boldsymbol{S}}^l \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{TT}}^l & \boldsymbol{K}_{\boldsymbol{TS}}^l \\ \boldsymbol{K}_{\boldsymbol{ST}}^l & \boldsymbol{K}_{\boldsymbol{SS}}^l \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{y}}_{\boldsymbol{T}}^l \\ \hat{\boldsymbol{y}}_{\boldsymbol{S}}^l \end{bmatrix} . \tag{5.52}$$

Subsequently, the stiffness matrices of some important coupling elements are presented. The situations and interface definitions are illustrated in *Tab.* 5.1. The interface definition drawings show which surfaces should be used for the interface, where the reference systems of the interfaces should be placed, how they should be oriented, and whether a stationary or moving interface should be used. The latter has an additional label $\boldsymbol{p}(s)$ at the origin of the path, and the path to be used is marked with a thick grey line. The specific path parameter s_0 to be used for the evaluation of the moving interfaces has to be found by projection of the target interface's location onto the path.

Usually, one is not interested in the local force distribution and deformation of the coupling elements, but in a matching of the overall behaviour between the mounting surfaces of the coupling element. Therefore, the geometry can be heavily simplified.

The stiffness values given by the supplier of the coupling components are usually measured (or calculated) for a load case with boundary conditions at the mounting surfaces of the coupling elements. Therefore, in order to use the supplier's values, the mounting surfaces of the coupling elements should be used as interfaces for simulation, where possible.

Table 5.1: Interface definitions for different types of links. The interfaces act on the orange marked surfaces with a centre of action at the coordinate system (u, v, w). Paths for moving interfaces are denoted with $\boldsymbol{p}(s)$.

	Drawing	**Interface definition**	
		Source interface	Target interface
Bearing			
Linear guide			
Ball screw			
Rack and pinion			
Gear			
Belt drive			

5.3.1 Bearings and Linear Guides

Bearings and linear guides are standard elements for the construction of moving axes. Both bearings and linear guides block the motion of some degrees of freedom and enable motion in the other directions. The blocked degrees of freedom can usually, as good approximation, be modelled as linear stiffness according to

$$\hat{\boldsymbol{f}}^{l}_{\boldsymbol{T}_d} = -\hat{\boldsymbol{f}}^{l}_{\boldsymbol{S}_d} = k_d \left(\hat{\boldsymbol{y}}^{l}_{\boldsymbol{T}_d} - \hat{\boldsymbol{y}}^{l}_{\boldsymbol{S}_d} \right) \qquad d \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\} \tag{5.53}$$

with the stiffness for any degree of freedom k_d. Thus follows, by comparison with *Eq.* (5.52), that the link stiffness components are diagonal matrices according to

$$\boldsymbol{K}^{l}_{\boldsymbol{TT}} = \boldsymbol{K}^{l}_{\boldsymbol{SS}} = -\boldsymbol{K}^{l}_{\boldsymbol{TS}} = -\boldsymbol{K}^{l}_{\boldsymbol{ST}} = \operatorname{diag}\left(k_u, k_v, k_w, k_{\varphi_u}, k_{\varphi_v}, k_{\varphi_w}\right) \ . \tag{5.54}$$

The meaning of the directional stiffness values for the examples of ball bearings and profiled linear guides is summarised in *Tab.* 5.2.

Table 5.2: Stiffness assignment for ball bearings and profiled linear guides

Direction	Ball bearings	Profiled linear guides
k_u	axial	0
k_v	radial	transversal
k_w	radial	normal
k_{φ_u}	0	roll
k_{φ_v}	tilt	pitch
k_{φ_w}	tilt	yaw

Because the coupling location of a bearing does not move relative to the bodies, the interfaces can both be modelled as stationary interfaces. In contrast, because the carriage of a linear guide moves along the rail, the latter has to be modelled as moving interface. As pointed out in *Section* 4.4, the surfaces used for moving interfaces must have an extent in all spatial directions. Therefore, for the rail of a linear guide, it is recommended to choose the side faces of the rail for the source interface.

5.3.2 Ball Screws

Ball screws allow the conversion of a rotary motion into a linear motion. A difference in the rotation between nut and spindle around the screw axis u leads to a translation between them both. The ratio of this conversion is defined by the pitch p of the screw, which gives displacement per revolution. Because the nut moves along the spindle, the spindle has to be modelled as moving interface. The mantle face of the spindle is appropriate as surface for the source interface. For the nut, the mounting faces can be selected.

The nut is moving force-free on the spindle if the rotation and translation between source and target interface are compatible with the screw pitch, i.e.

$$\left(\hat{\boldsymbol{y}}^{l}_{\boldsymbol{T}_u} - \hat{\boldsymbol{y}}^{l}_{\boldsymbol{S}_u} \right) = \frac{p}{2\pi} \left(\hat{\boldsymbol{y}}^{l}_{\boldsymbol{T}_{\varphi u}} - \hat{\boldsymbol{y}}^{l}_{\boldsymbol{S}_{\varphi u}} \right) \ . \tag{5.55}$$

A deviation from this leads to an elastic force on the nut, and in opposite direction, on the spindle according to

$$\hat{\boldsymbol{f}}^{l}_{\boldsymbol{T}_u} = -\hat{\boldsymbol{f}}^{l}_{\boldsymbol{S}_u} = k \left(\hat{\boldsymbol{y}}^{l}_{\boldsymbol{T}_u} - \hat{\boldsymbol{y}}^{l}_{\boldsymbol{S}_u} \right) - k \frac{p}{2\pi} \left(\hat{\boldsymbol{y}}^{l}_{\boldsymbol{T}_{\varphi u}} - \hat{\boldsymbol{y}}^{l}_{\boldsymbol{S}_{\varphi u}} \right) \ , \tag{5.56}$$

where k is the axial stiffness of the nut. With this force, also a torque is generated according to

$$\hat{f}^{l}_{\boldsymbol{T}_{\varphi u}} = -\hat{f}^{l}_{\boldsymbol{S}_{\varphi u}} = \frac{p}{2\pi}\,\hat{f}^{l}_{\boldsymbol{T}_{u}} \,. \tag{5.57}$$

These equations can be put together to the link stiffness components

$$\boldsymbol{K}^{l}_{\boldsymbol{TT}} = \boldsymbol{K}^{l}_{\boldsymbol{SS}} = -\boldsymbol{K}^{l}_{\boldsymbol{TS}} = -\boldsymbol{K}^{l}_{\boldsymbol{ST}} = \begin{bmatrix} k & 0 & 0 & -k\,\frac{p}{2\pi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -k\,\frac{p}{2\pi} & 0 & 0 & k\,\left(\frac{p}{2\pi}\right)^{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \,. \tag{5.58}$$

Additionally, radial and tilting stiffness could be used equally as with bearings.

5.3.3 Rack and Pinion

Similarly to ball screws, rack and pinion systems allow the transformation of a translation into a rotation and vice versa. The rack has to be modelled as moving interface, where the pinion can be modelled as stationary interface. For the rack, it is proposed to use both side faces plus the top face for the interface, again because faces for moving interfaces have to have an extent in all directions.

A force on a tooth leads to a force normal to the tooth flank, which is inclined at the profile angle of the gear. This results in tangential and a radial force component. The tangential force component is generating torque at the pinion, whereas the radial force component pushes the rack and the pinion apart. Because the direction of the radial force is independent of the direction of the torque, this force is non-linear and can thus not be modelled using a stiffness matrix. Therefore, here only the tangential force is considered and the non-linear force component, if desired, has to be modelled separately for transient simulation.

If the path for the moving interface is placed along the pitch line, for the rack interface, a force on a tooth leads to a force only and no torque is induced. The same force applied on the pinion, however, leads to a force and a torque on the pinion's interface because of the cantilever with the pinion radius r.

This leads to the stiffness matrix components

$$\boldsymbol{K}^{l}_{\boldsymbol{TT}} = \begin{bmatrix} k & 0 & 0 & 0 & -k\,r & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -k\,r & 0 & 0 & 0 & k\,r^{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} , \quad \boldsymbol{K}^{l}_{\boldsymbol{TS}} = \begin{bmatrix} -k & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ k\,r & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.59}$$

$$\boldsymbol{K}^{l}_{\boldsymbol{ST}} = \begin{bmatrix} -k & 0 & 0 & 0 & k\,r & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} , \quad \boldsymbol{K}^{l}_{\boldsymbol{SS}} = \begin{bmatrix} k & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \,. \tag{5.60}$$

5.3.4 Gears

Gear drives are used to translate a rotational motion into another rotational motion by a specific gear ratio, defined by the wheel radii r_T and r_S of the target and source interface, respectively.

A displacement between the teeth in contact leads to a force proportional to the tooth stiffness k. The displacements of the single teeth depend on the rotation of the gears and the radii. Moreover, the torque reaction is proportional to the radius and the tooth force. This leads to the stiffness matrix components

$$\boldsymbol{K}^l_{\boldsymbol{TT}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k & -k\,r_T & 0 & 0 \\ 0 & 0 & -k\,r_T & k\,r_T^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{K}^l_{\boldsymbol{TS}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -k & -k\,r_S & 0 & 0 \\ 0 & 0 & k\,r_T & k\,r_T\,r_S & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.61}$$

$$\boldsymbol{K}^l_{\boldsymbol{ST}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -k & k\,r_T & 0 & 0 \\ 0 & 0 & -k\,r_S & k\,r_S\,r_T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{K}^l_{\boldsymbol{SS}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k & k\,r_S & 0 & 0 \\ 0 & 0 & k\,r_S & k\,r_S^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.62}$$

The interfaces for the gears can be defined at the circumference of the wheels as stationary interfaces with centre of action at the rotation centre.

Again, the non-linear radial forces are omitted here and have to be modelled separately, if desired.

5.3.5 Belt Drives

Belt drives are similar to gear drives, however, because the belts are prestressed, a torque on the system leads to an increase in force on one belt side and a decrease in the other. In this operation point, the behaviour is as if one side of the belt was pulling and the other side was pushing. Therefore, force pairs arise which compensate for a resulting force on the wheels.

The stiffness matrix components for a belt drive with wheel radii r_T and r_S and the stiffness of one belt strand k reads

$$\boldsymbol{K}^l_{\boldsymbol{TT}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\,k\,r_T^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{K}^l_{\boldsymbol{TS}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2\,k\,r_T\,r_S & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.63}$$

$$\boldsymbol{K}^l_{\boldsymbol{ST}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2\,k\,r_S\,r_T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{K}^l_{\boldsymbol{SS}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\,k\,r_S^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.64}$$

The prestress force is relevant for the static deformation only and can usually be neglected for dynamics analyses. If one wants to model the prestress effect, however, a force input with the link input matrix $\hat{\boldsymbol{B}}^l$ can be defined.

5.4 Discussion of Flexible Multi-Body Simulation

A straightforward procedure has been derived, which allows the assembly of system and interface matrices for a rotated body by means of the original system matrices, a reduction basis, and explicitly defined rigid body modes. For the sake of simplicity, and because machine tools dynamics usually can be analysed by means of linear analyses at a specific working point, a derivation of the non-linear system mass and interface matrices for transient simulation with large rotation has been suspended. The advantage of this is that the system is described in the standard second-order form for linear dynamic systems and the well-known analysis methods for this kind of systems, like static, modal, and frequency response analyses can be performed without loss of information. Transient simulation can be performed with the structure linearised for any specific orientation and location of the bodies. Despite the linearised structure, non-linear components like load-dependent stiffness or non-linear controllers can be used for transient simulation with the interfaces of the structure.

The drawback of the linearisation is that transient simulation involving large rotations is not possible. Although this is not required for many analyses with machine tools, it could be of relevance for special cases or for other fields of application, like robotics or mechanisms. Therefore, the formulation and implementation of the equations of motion by means of the available model data is picked up again in the outlook.

The composition of system matrices of flexible bodies to a linear flexible multi-body system has been presented. Especially the coupling of the bodies by means of linear stiffness and viscous damping has been discussed. The derivation of link stiffness matrices for some important coupling elements in mechanical engineering has been presented, what demonstrates the effectiveness of the proposed formulation of stationary and moving interfaces. The same procedure can be used for the derivation of arbitrary couplings between multiple bodies, as long as the force law is linear. For non-linear couplings, the interfaces can be utilised to apply the non-linear force law during transient simulation.

6 Software Realisation

A new software package has been developed which implements the methods for model order reduction, interface modelling, and composition of flexible bodies as presented in the preceding chapters. The software tool named MORe, what is an acronym for Model Order Reduction and more, is presented in this chapter.

The ambition of MORe is not only to implement the presented simulation methods but also to embed the functionalities in an efficient and comprehensible work-flow for the simulation of machine tools, especially for accompanying the time-critical design phase of new products.

In the following section, some information about the implementation and software packages used is given. Subsequently, in *Section* 6.2, the structure of MORe models is described and the modelling work-flow is presented in *Section* 6.3. Lastly, an overview of the implemented analysis types is given in *Section* 6.4.

6.1 Overview of the Implementation

The major part of MORe is written with the object oriented and interpreted programming language Python [81]. The Traits [108] package is used in order to facilitate ensuring data integrity. The Traits package allows the specification of data type verification for object attributes as well as event management.

MORe comprises a full-featured application programming interface (API), but for an efficient work-flow, a graphical user interface (GUI) is inevitable. Strongly related to the Traits package, the TraitsUI [109] package provides functionalities for creation of basic GUIs. With the TraitsUI package, a comprehensive yet flexible GUI has been developed which assists the user throughout the modelling and analysis process with MORe.

Working with finite element models and meshes involves handling large amounts of data. This requires efficient tools for mathematical calculations and, especially for model order reduction, linear algebra algorithms. SciPy [52] offers a vast collection of state-of-the-art algorithms for versatile fields of scientific data manipulation. MORe makes use of data types, eigenvalue solvers, and linear equation solvers for both dense and sparse matrices provided by SciPy. The utilised eigenvalue solver is the implicitly restarted Arnoldi method from ARPACK [64] and for the solution of large-scale sparse systems of equations, the LU-decomposition SuperLU [67] is used. Both ARPACK and SuperLU are interfaced by SciPy.

Three-dimensional visualisation of the models is indispensable for an efficient modelling process as well as for the interpretation of simulation results. The visualisation of model geometry is based on the responsive and visual appealing rendering functionalities featured by the Mayavi [82] package.

One of the most commonly used tools for structural finite element simulation of machine tools undoubtedly is ANSYS Mechanical [4]. Because ANSYS Mechanical provides robust functionalities for geometry import and meshing, it is used as a preprocessor for MORe. An interface to ANSYS Mechanical has been developed, which features the export of meshed bodies by means of an ANSYS Mechanical macro (JScript). This macro handles the geometry information, i.e. the interconnection between vertices, faces, and bodies, and invokes further ANSYS Parametric Design Language (APDL) macros that handle all information about the finite element and mesh definition, i.e. nodes, element connectivity, system matrices, DOF mapping, and algebraic constraints.

For state-of-the-art system simulation and data analysis, MATLAB [72] and Simulink [100] are the tools of choice. A MATLAB class library has been crated which allows the import of MORe projects into the MATLAB workspace. The object oriented data structure of the model hierarchy used by MORe in Python is reflected with MATLAB classes. This enables the user to analyse and modify model properties directly with MATLAB, however, saving the modifications to the original project has not been implemented. A Simulink block library allows a straightforward integration of flexible multi-body systems with controllers and drives by means of graphical programming using block diagrams.

6.2 Model Hierarchy

A MORe model consists of multiple bodies, denoted as components. Interfaces to the components specify the application of loads and the readout of displacements and rotations. The components are assembled to a composition and interconnected using links between the interfaces. A schematic example of a MORe model is shown in *Fig.* 6.1, and subsequently, these basic elements of the models are described.

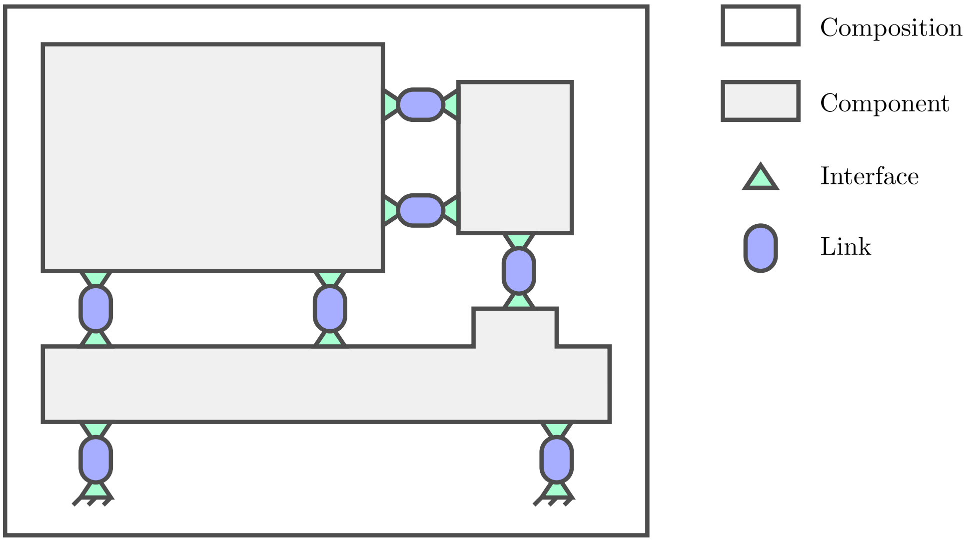


Figure 6.1: Schematic representation of a MORe model

Component The basis for a MORe model are finite element models of single components, e.g. axes or machine basis. The components are to be free in space, i.e. they have exactly six rigid body modes each. Thus, contacts between parts of a component have to be modelled with the finite element software in advance, so that a component is then one unmounted body in space.

A component in MORe handles all the component-related properties, i.e.

- geometry and topology (vertices, edges, faces, bodies),
- finite element mesh (nodes, elements),
- original system,
- reduced systems,
- rigid body system, and
- interfaces.

From one component, multiple reduced order systems can be derived, e.g. for different frequency ranges or different accuracies. Furthermore, a rigid body system can be derived, which allows a rigid multi-body simulation. The system to use for simulation can be specified as default system.

Interface Inputs and outputs to the system are handled equally and are implemented according to *Chapter* 4. A stationary interface is defined by one or multiple faces and a local reference system, and a moving interfaces is defined by faces and a path.

A face is specified by a topological element with a persistent identification number defined by the finite element software. Thus, as long as the assignment of identification numbers does not change in the finite element software, the assignment of the faces to an interface is retained. This allows modifying the mesh or even parts of the geometry of a finite element model and reimporting it without the need of respecifying all interfaces. In order to simplify the geometrical restriction of the action of an interface, limit distances to the location of an interface can be set. Thus, limiting the action to a fraction of a face does not require a geometrical feature to be created.

The location of a local reference system is specified by the centre of a set of faces and can additionally be offset in all directions. The orientation is either given by face normals, edges and vertices, or by vectors described in the body reference system.

A path is specified either using a start- and end-point, defined through faces, edges, or vertices, or using an edge and a start vertex. The latter allows the definition of curved paths, realised using straight segments created by the finite elements on the edge.

Link Connections between two interfaces or between one interface and the inertial system are defined using links as described in *Section* 5.2.2.

A link is defined by a source interface, a target interface, and optionally a link properties item. Link properties are defined in a catalogue on composition level and can be reused for multiple links. Changing a parameter for a specific coupling element, e.g. varying the stiffness of one type of linear guides, is therefore managed centrally and has not to be repeated for each instance of this coupling type. Furthermore, links can be used for displacement readout and load application.

For both the location and orientation of a link, a master can be defined. If a master interface is defined, the slave interface location or orientation is then changed to the one of the master. This allows to ensure an exact matching of the locations for both interfaces of a link.

Composition All information about the composed model is handled by the composition object. The fundamental properties are

- components,
- links,
- link properties, and
- kinematics definition.

The kinematics section allows the specification of moving axes consisting of one or multiple components. The moving direction or the rotation axis are defined using local coordinate systems of available interfaces, e.g. a rotation axis for a spindle is defined using the local reference system of a bearing interface. Using the kinematics section, arbitrary positions in the workspace can be set for subsequent analyses. The orientation of the component's system matrices and the update of links with moving interfaces is accomplished automatically.

6.3 Modelling Work-Flow

Special attention has been paid to a straightforward work-flow for the modelling and simulation processes. The toolchain for MORe is shown in *Fig.* 6.2. MORe benefits from the strengths of the finite element software ANSYS Mechanical and the data processing and simulation tools MATLAB and Simulink, but complements those tools with functionalities taylored for machine tool simulation.

The modelling work-flow is outlined in *Fig.* 6.3. First, finite element models of the single components are created and exported by means of a predefined macro. At this stage, no information about interfaces

Figure 6.2: Toolchain for modelling and simulation with MORe

or the final application have to be provided. Subsequently, the components are imported to MORe, the interfaces are defined, and the model order reduction is performed. At this stage, it is only important to know where the interfaces act, the links and link properties, however, are not yet to be defined. Afterwards, the composition definition is finalised with the definition of link properties, links and kinematics.

In order to check the model for plausibility, modal, frequency response, and static structural analyses can be conducted at this stage.

Finally, the project file can directly be loaded with MATLAB and a Simulink model can be created using predefined blocks for composition, inputs, outputs, and result data handling. Both interfaces and links can be used as inputs or outputs. Furthermore, acceleration inputs, either to components or to the composition, are also provided for the simulation of, e.g. gravitational load cases.

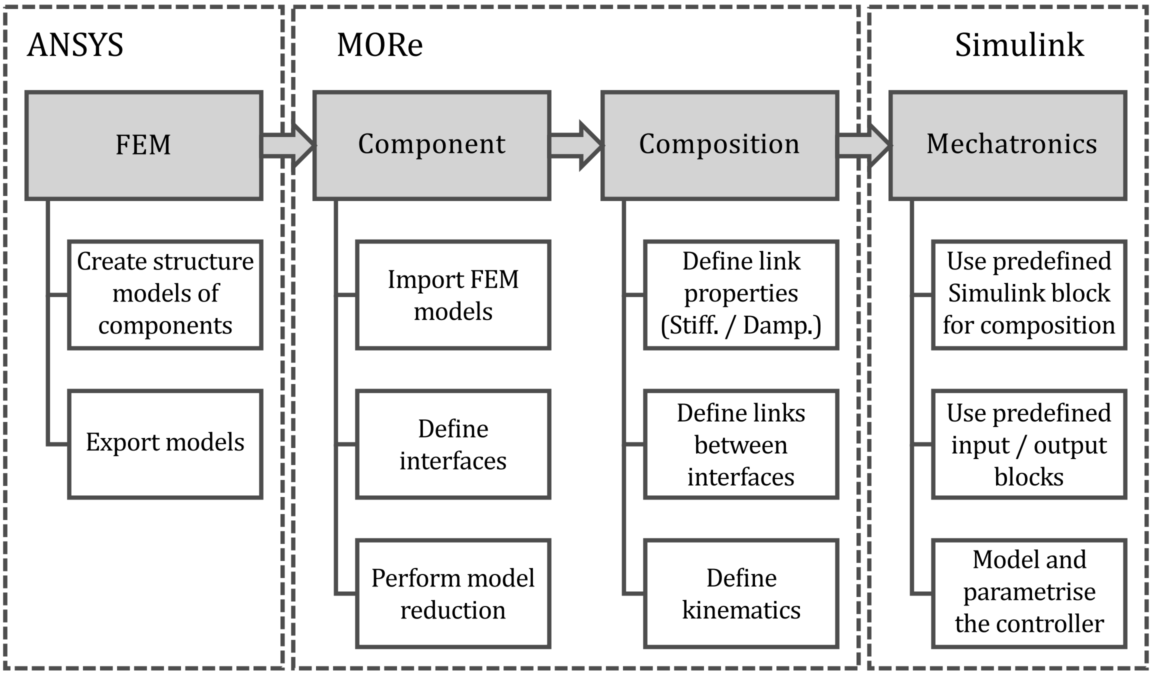


Figure 6.3: Modelling work-flow with MORe

6.3.1 Modification Work-Flow

During the modelling process or even after finalisation of a model, often modifications to parts of the model are required, e.g. due to modelling errors, model detail enhancement, or changes in the design. It is important to keep the changes isolated from the parts of the model which are not directly involved. The goal is to keep all model information which is still valid after a modification, and thus, to minimise the number of repetitions for the same task. Subsequently, the necessary work is listed for various modification cases.

Geometry Modifications of the geometry cause that the finite element model has to be updated. Owing to the interfacing of ANSYS with current computer aided design (CAD) tools, it is possible to modify the geometry without loosing the assignment to the topological features that do not change. As a consequence, after importing the updated finite element model to MORe, the definitions of interfaces remain valid. Of course this is not possible in the case that the involved topological features do not exist anymore, e.g. because an involved face has been removed.

Because the finite element model has changed, the model order reduction has to be repeated for the modified component.

Finite Element Model If the finite element model changes, e.g. due to modification of the mesh, material properties, or contacts, then the model has to be reimported to MORe. Because the geometry has not changed, the interface definitions are not affected by this. The model order reduction, however, has to be performed again for this particular component.

Interface If any interface changes, the model order reduction has to be repeated in order to ensure the accuracy of the affected transfer functions. However, all the desired changes can be undertaken without any relevant waiting time and the time-consuming tasks can be postponed.

Link Creation, deletion, or modification of links does not affect any component and can thus be performed with a minimal effort. Only the assembly of the composition's system matrices has to be repeated. This, however, is not a time-consuming task.

Link Property Modification of link properties does also not affect the reduced models of components. Therefore, changes in the coupling parameters can be analysed efficiently without the need of reducing models again.

Kinematics Definition and Axis Positions Thanks to the moving interfaces, changes in the position of components can be modelled without affecting any interface's definition, and thanks to the reorientation of system matrices, the same holds for the rotation of components.

Controller Modifications to the Simulink model does not affect the MORe composition at all. Therefore, the controller, either parameters or the controller structure, can be modified repeatedly without the need of working on the definition of the MORe model.

This listing shows the effectiveness of the work-flow and model hierarchy chosen for MORe. Neither a time-consuming computation nor a cumbersome manual task have to be repeated, if it is not logically required.

6.4 Featured Analyses

A composition contains all necessary information for versatile simulative analyses. The most important analyses of the composition, i.e. the structure without controllers and drives, are implemented within the GUI of MORe. Controller analysis and further custom analyses can be performed by means of MATLAB and Simulink. The predefined analyses are listed in *Tab.* 6.1 with information about the supported environment and animation capabilities. Based on the data from these basic analyses, other analyses can be conducted using the comprehensive functionalities of MATLAB and Simulink or Python.

Analyses of the variation of properties within the workspace are called spatial analyses. A spatial static analysis can, e.g. be used to determine geometric accuracy of a machine tool under static loads like gravitation or disturbance forces. Spatial FRF analysis allows the analysis of the dynamic performance of a machine tool within the range of travel of the axes.

The animation of simulation results is crucial for their interpretation. Therefore, following animation options have been implemented:

- transient animation,
- shape animation for modal, static, and FRF analysis results, and

- spatial static results animation.

All animation options feature scaling of the deformation, video rendering, and plotting or tabular listing of link trajectories.

Table 6.1: Analyses performable by means of MORe or a MORe model in MATLAB and Simulink

Analysis type	**MORe**[1)]	**MATLAB**[2)]	**Animation**
Static	●	●	●
Spatial static	●	●	●
FRF	●	●	●
Spatial FRF		●	
Modal	●	●	●
Transient with linearised structure		●	●

[1)] mechanical structure only, [2)] open- and closed-loop

6.5 Discussion of the Software Implementation

MORe is a comprehensive software package that allows an efficient and straightforward work-flow and features a multitude of analysis options. Due to the accessibility of model data throughout MORe and MATLAB, an effective analysis of properties and parameters is possible. The access to all simulation data for custom analyses and the animation features help the user to establish a detailed understanding of the machine's structure and its interaction with the controllers and drives.

Because of the open architecture of the software and because of its consistent API, MORe is a convenient tool for further development and analysis of topics in the field of model order reduction or flexible multi-body simulation.

7 Design to Specifications

After having introduced modelling methods and the modelling software MORe, the question arises, how to integrate simulation effectively into the design process of a machine tool.

This chapter describes a newly developed design process named Design to Specifications (DtS) that starts with the derivation of requirements on the structure of a machine tool such that the specifications on the productivity and accuracy of the complete system can be fulfilled. This information can then be used to find parameters for the design like, e.g. fundamental dimensions, bearing or guide distances, or stiffness values for structural components by basic calculations. The actual dynamic performance of a particular design is then evaluated by means of simulation with a MORe model.

Here, the trajectory-following accuracy for dynamically moving axes is examined exclusively. Process forces and process stability issues are not covered here. However, because manufacturing processes are usually split into a roughing and a finishing process, and because the latter is nearly force-free, the assumption of negligible process forces for the analysis of contouring accuracy is applicable to a wide area of machine tool applications.

In *Fig.* 7.1, the DtS process is shown as a flow-chart. The yellow block, concerning the modelling of the system, is covered in *Chapter* 6, the green blocks, concerning the derivation of requirements, are described in *Section* 7.1 and the blue block, concerning the verification of the specifications using a MORe model, is discussed in *Section* 7.2.

7.1 Requirements on the Structure

In this section, the definition of specifications and the derivation of requirements are explained. The presented procedure enables the engineer to consecutively derive requirements on

- the dynamic limits (jerk or acceleration limit) out of productivity specifications,
- the required controller bandwidth out of the dynamic limits and the accuracy specification, and
- the required critical eigenfrequency out of the controller bandwidth,

which lead to a machine that complies with the specifications.

7.1.1 Specifications

The dynamic capabilities of a machine tool shall be specified by means of the productivity and accuracy.

Productivity The productivity can be expressed as the time needed to process a specific contour. This can be, e.g. the machining time for a reference part. Alternatively, if the dynamic limits, i.e. the jerk and acceleration limits, are known a priori, they can be used directly as dynamics specifications.

Accuracy The accuracy can be specified as a maximum contour error $\hat{e}_c$ at the TCP which represents the maximum normal distance of the actual to the desired path. In order to quantify the accuracy in an axis-wise manner, Spescha, Weikert, Zirn, and Wegener [103] introduced the dynamic tracking error d (see *Definition* 7.2.1). The dynamic tracking error is the part of the tracking error, which cannot be compensated using a lag and is thus the portion of the tracking error which is relevant for synchronised axes.

The contour error which can appear in the worst case is a combination of the maximum dynamic

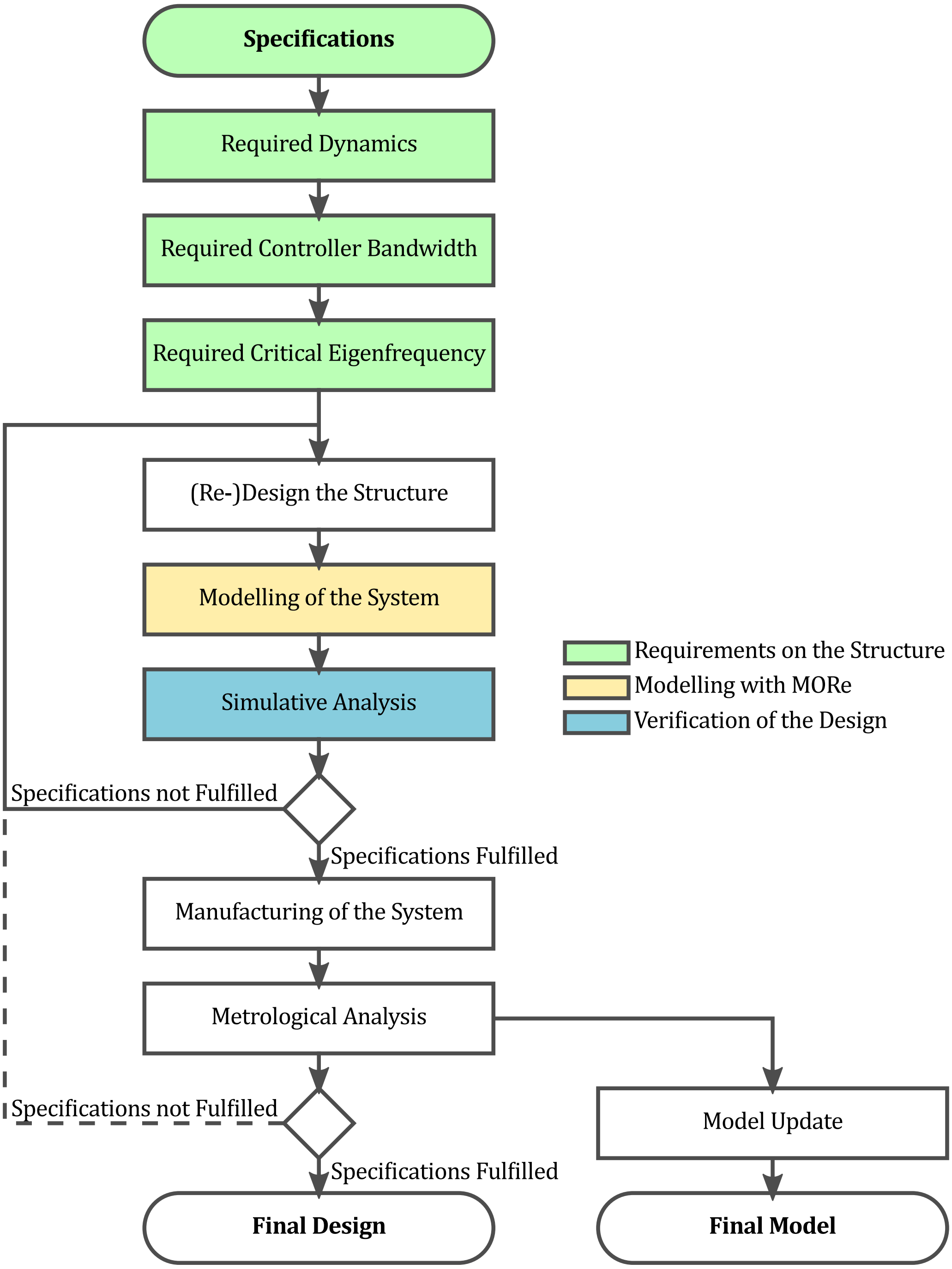


Figure 7.1: Flow-chart of the Design to Specifications process

tracking error on all axes. For a Cartesian arrangement of n_a axes, each with a maximum dynamic tracking error $\hat{d}_i$, the worst case contour error is the Euclidean distance out of all dynamic tracking errors as

$$\hat{e}_c = \sqrt{\sum_{i=1}^{n_a} \hat{d}_i^2} \; . \tag{7.1}$$

Trajectories which lead to a constructive superposition of the dynamic tracking errors are very rare. They must contain steps from positive to negative maximum acceleration or jerk on all axes at the same instant. Therefore, it is recommended to use the maximum dynamic tracking error of the single axes for the accuracy judgement.

For more information about the concepts of synchronisation and dynamic tracking error of feed axes, the reader is referred to Spescha et al. [103].

7.1.2 Required Dynamics

If the jerk or acceleration limit is known from the specifications, this section is obsolete. Otherwise, the required dynamic limits have to be derived from the productivity specification. Therefore, a (possibly virtual) computer numerical control (CNC) kernel can be used, which allows planning the reference trajectory with defined acceleration and jerk limits $\hat{a}$ and $\hat{j}$, respectively. With a few attempts, the dynamic limits leading to the desired productivity may be found. In this step, only the trajectory planning and kinematics, but no dynamic properties or accuracy limits of the machine are taken into account.

7.1.3 Required Controller Bandwidth

In this section, the relationship between controller bandwidth, accuracy, and acceleration or jerk limits is presented. Therefore, a simplified feed axis model is introduced and the worst-case error for acceleration- and jerk-limited trajectories is outlined. The verification of the error bounds using a MORe model is described later in *Section* 7.2.

7.1.3.1 Simplified Transfer Function

Spescha et al. [103] derived an upper bound for the contour error for acceleration- and jerk-limited trajectories for feed axes with known bandwidth and damping of the position controller. Therefore, the closed position loop of the system was simplified to a second order low-pass system with bandwidth ω_c and damping D_c with the transfer function

$$G(s) = \frac{\omega_c^2}{s^2 + 2\, D_c\, \omega_c\, s + \omega_c^2} \; . \tag{7.2}$$

This is a simplification, but for every feed axis, a controller parametrisation can be found so that the transfer function is similar to the transfer function in *Eq.* (7.2).

Further, Spescha et al. [103] showed that the optimal damping ratio for the position controller is

$$D_c = \frac{1}{\sqrt{2}} \; . \tag{7.3}$$

Therefore, this damping ratio will be used in the following.

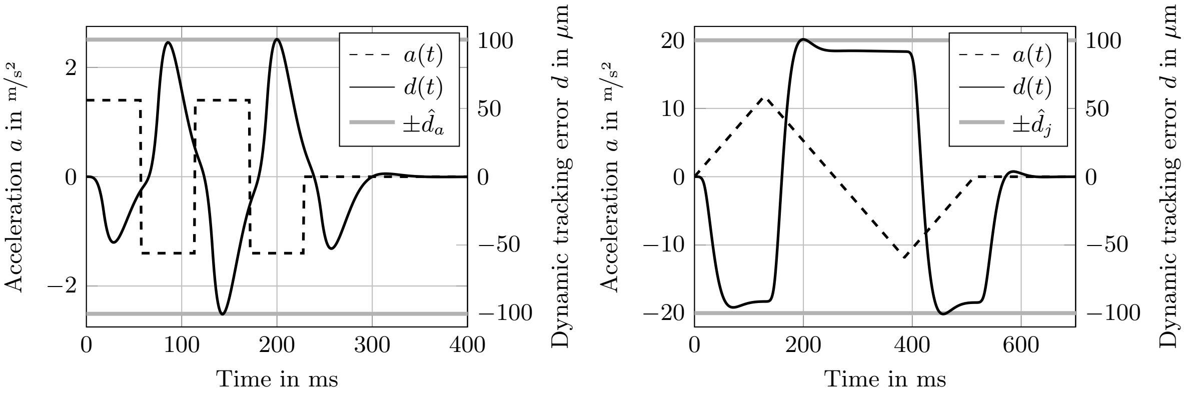
7.1.3.2 Dynamic Tracking Error Bound

For acceleration-limited trajectories, the trajectory leading to the maximum dynamic tracking error for an axis is a consecutive switching from maximum positive to maximum negative acceleration with the appropriate acceleration time as shown in *Fig.* 7.2(a). The maximum dynamic tracking error, i.e. the upper bound for the dynamic tracking error for acceleration-limited trajectories, can be shown to satisfy

$$\hat{d}_a = 2\left(e^{\frac{-\pi}{2}} + e^{\frac{-3\pi}{2}}\right)\frac{\hat{a}}{\omega_c^2}\,. \tag{7.4}$$

Similarly, for jerk-limited trajectories, there is a worst case trajectory too. This is a consecutive switching from maximum positive to maximum negative jerk according to *Fig.* 7.2(b). The upper limit for the dynamic tracking error for jerk-limited trajectories is

$$\hat{d}_j = \left(\frac{\sqrt{2}}{3} + 2\,e^{\frac{-5\pi}{4}}\right)\frac{\hat{j}}{\omega_c^3}\,. \tag{7.5}$$



(a) Acceleration-limited profile ($\hat{a}$ = 1.4 m/s^2)

(b) Jerk-limited profile ($\hat{j}$ = 92 m/s^3)

Figure 7.2: Acceleration and dynamic tracking error for the worst-case trajectory for an example system with bandwidth ω_c = 78 rad/s and damping ratio $D_c = 1/\sqrt{2}$

7.1.3.3 Required Controller Bandwidth

Since the interconnection of accuracy, dynamic limits, and controller bandwidth is known from *Eq.* (7.4) and *Eq.* (7.5), the required controller bandwidth can be calculated for acceleration-limited trajectories according to

$$\omega_{c_a} \geq \left(2\left(e^{\frac{-\pi}{2}} + e^{\frac{-3\pi}{2}}\right)\frac{\hat{a}}{\hat{d}_a}\right)^{\frac{1}{2}} \tag{7.6}$$

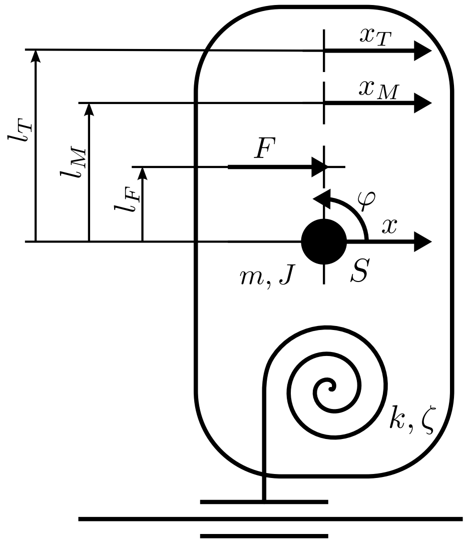
and for jerk-limited trajectories according to

$$\omega_{c_j} \geq \left(\left(\frac{\sqrt{2}}{3} + 2\,e^{\frac{-5\pi}{4}}\right)\frac{\hat{j}}{\hat{d}_j}\right)^{\frac{1}{3}}\,. \tag{7.7}$$

7.1.4 Required Critical Eigenfrequency

In this section, an estimation for the critical eigenfrequency of an axis is described. If the eigenfrequency is known, requirements on mass, moment of inertia, and stiffness of structural components can be derived accordingly.

7.1.4.1 Characterisation and Parametrisation of the Mechanical System



m	kg	mass
J	$\mathrm{kg\,m^2}$	moment of inertia
k	Nm/rad	rotational stiffness
ζ	Nm s/rad	damping coefficient
F	N	drive force
x	m	translatory state
φ	rad	rotary state
x_M	m	encoder signal
x_T	m	TCP position
l_F	m	offset from S to force action
l_M	m	offset from S encoder
l_T	m	offset from S to TCP

Figure 7.3: Schematic model of an elastically mounted rigid body

For the characterisation of the behaviour of a mechanical axis, a rigid body model according to *Fig.* 7.3 is used. This is a simplification for the first tilting mode of a deformable axis, but the dominant effects can be shown using this model. Further cases of elastically mounted and actuated rigid bodies were listed by Zirn [119].

Mathematically, this system can be written in matrix form using a system of second order linear ordinary differential equations as

$$\boldsymbol{M}\,\ddot{\boldsymbol{x}} + \boldsymbol{D}\,\dot{\boldsymbol{x}} + \boldsymbol{K}\,\boldsymbol{x} = \boldsymbol{B}\,F \tag{7.8}$$

$$\boldsymbol{y} = \boldsymbol{C}\,x \tag{7.9}$$

with the state vector $\boldsymbol{x}$ and the output vector $\boldsymbol{y}$ according to

$$\boldsymbol{x} = \begin{bmatrix} x \\ \varphi \end{bmatrix} \; , \quad \boldsymbol{y} = \begin{bmatrix} x_M \\ x_T \end{bmatrix} \tag{7.10}$$

and the system matrices according to

$$\boldsymbol{M} = \begin{bmatrix} m & 0 \\ 0 & J \end{bmatrix} \; , \quad \boldsymbol{D} = \begin{bmatrix} 0 & 0 \\ 0 & \zeta \end{bmatrix} \; , \quad \boldsymbol{K} = \begin{bmatrix} 0 & 0 \\ 0 & k \end{bmatrix} \; , \quad \boldsymbol{B} = \begin{bmatrix} 1 \\ l_F \end{bmatrix} \; , \quad \boldsymbol{C} = \begin{bmatrix} 1 & l_M \\ 1 & l_T \end{bmatrix} \; . \tag{7.11}$$

The transfer function can be calculated in matrix form as

$$\boldsymbol{G}(s) = \boldsymbol{C}\,\left(\boldsymbol{M}\,s^2 + \boldsymbol{D}\,s + \boldsymbol{K}\right)^{-1}\,\boldsymbol{B}\,, \tag{7.12}$$

what leads to the plant transfer function for the velocity at the encoder

$$G_M(s) = \frac{1}{m\,s}\,\frac{(m\,l_F\,l_M + J)\,s^2 + \zeta\,s + k}{J\,s^2 + \zeta\,s + k} \tag{7.13}$$

and, analogously, at the TCP

$$G_T(s) = \frac{1}{m\,s}\,\frac{(m\,l_F\,l_T + J)\,s^2 + \zeta\,s + k}{J\,s^2 + \zeta\,s + k}\;. \tag{7.14}$$

Because structural damping is usually small for mechanical components and for the sake of simplicity, the damping coefficient is omitted ($\zeta = 0$). This leads to the simplified transfer functions

$$G_M(s) = \frac{1}{m\,s}\,\frac{(m\,l_F\,l_M + J)\,s^2 + k}{J\,s^2 + k} \tag{7.15}$$

and for the TCP

$$G_T(s) = \frac{1}{m\,s}\,\frac{(m\,l_F\,l_T + J)\,s^2 + k}{J\,s^2 + k}\;. \tag{7.16}$$

From these equations, the resonance frequency ω_P, the antiresonance frequency at the encoder ω_M, and the antiresonance frequency at the TCP ω_T can directly be identified to

$$\begin{align} \omega_P &= \sqrt{\frac{k}{J}} \tag{7.17}\\ \omega_M &= \sqrt{\frac{k}{m\,l_F\,l_M + J}} \tag{7.18}\\ \omega_T &= \sqrt{\frac{k}{m\,l_F\,l_T + J}} \tag{7.19} \end{align}$$

by setting $s = i\omega$ and calculating the poles and zeros.

Depending on the sign and dimension of the geometrical offsets, different cases can arise, which are briefly described in *Tab.* 7.1 for the encoder and in *Tab.* 7.2 for the TCP. Example frequency responses are illustrated in *Fig.* 7.4.

Table 7.1: Distinction of cases for different force and encoder offsets

1)	$\mathrm{sign}(l_F) = \mathrm{sign}(l_M)$ The antiresonance frequency lies below the resonance frequency ($\omega_M < \omega_P$). From a control theory point of view, this case is to prefer because the phase rises between ω_M and ω_P, leading to increased stability margins (see *Fig.* 7.4, case 1).
2)	$\mathrm{sign}(l_F) \neq \mathrm{sign}(l_M)$ and $J > m\,l_F\,l_M$ The antiresonance frequency lies above the resonance frequency ($\omega_M > \omega_P$). From a control theory point of view, this case has to be avoided, because the phase drops below -180° between ω_P and ω_M (see *Fig.* 7.4, case 2). This leads to stability issues with a cascaded loop controller.
3)	$\mathrm{sign}(l_F) \neq \mathrm{sign}(l_M)$ and $J < m\,l_F\,l_M$ No antiresonance arises, the zero of *Eq.* (7.15) has a positive real part and, therefore, the system is not of minimal phase. From a control theory point of view, this case has to be avoided, because the phase drops below -180° after ω_P (see *Fig.* 7.4, case 3). This leads to stability issues with a cascaded loop controller.

As a conclusion, in order to avoid stability issues with a cascaded loop controller, the offsets l_M and l_F have to have the same sign, i.e. the encoder has to be on the same side of the centre of mass as the motor.

Table 7.2: Distinction of cases for different force and TCP offsets

A)	$\mathrm{sign}(l_F) = \mathrm{sign}(l_T)$ The antiresonance frequency lies below the resonance frequency ($\omega_T < \omega_P$).
B)	$\mathrm{sign}(l_F) \neq \mathrm{sign}(l_T)$ and $J > m\, l_F\, l_T$ The antiresonance frequency lies above the resonance frequency ($\omega_T > \omega_P$).
C)	$\mathrm{sign}(l_F) \neq \mathrm{sign}(l_T)$ and $J < m\, l_F\, l_T$ No antiresonance arises, the zero of *Eq.* (7.16) has a positive real part and therefore, the system is not of minimal phase.

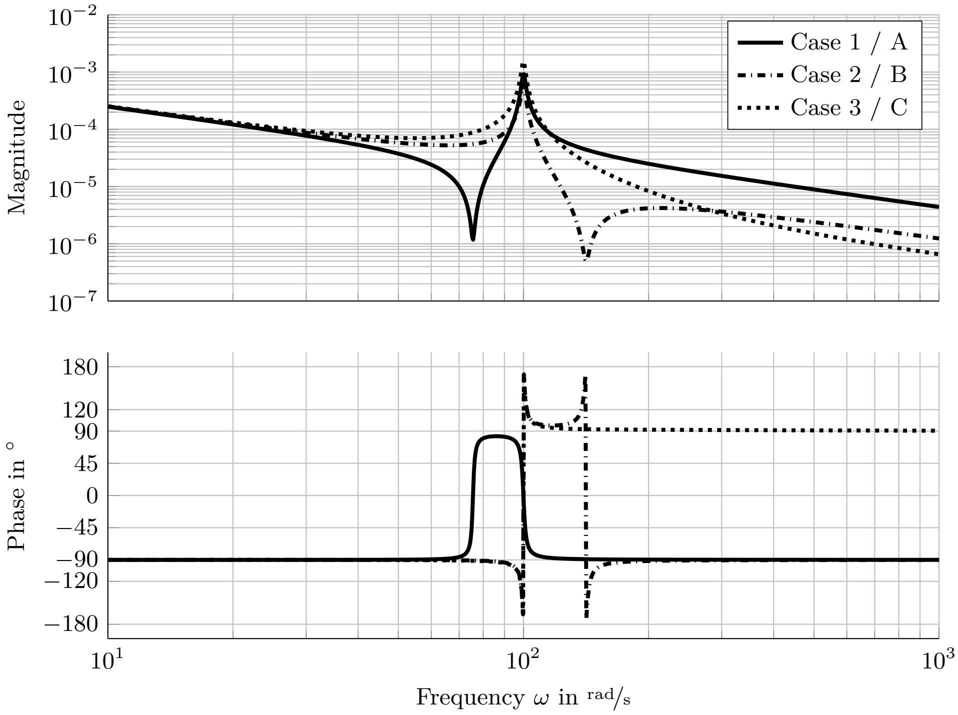


Figure 7.4: Comparison of cases 1 to 3 from *Tab.* 7.1 and cases A to C from *Tab.* 7.2, respectively ($m = 400\,\mathrm{kg}$, $J = 4\,\mathrm{kg\,m^2}$, $k = 40\,\mathrm{Nm/mrad}$, $\zeta = 8\,\mathrm{Nm\,s/rad}$, $l_F = 0.05\,\mathrm{m}$, case 1: $l_M = 0.15\,\mathrm{m}$, case 2: $l_M = -0.1\,\mathrm{m}$, case 3: $l_M = -0.25\,\mathrm{m}$)

7.1.4.2 Controller Parametrisation for Optimal Damping of Structural Modes

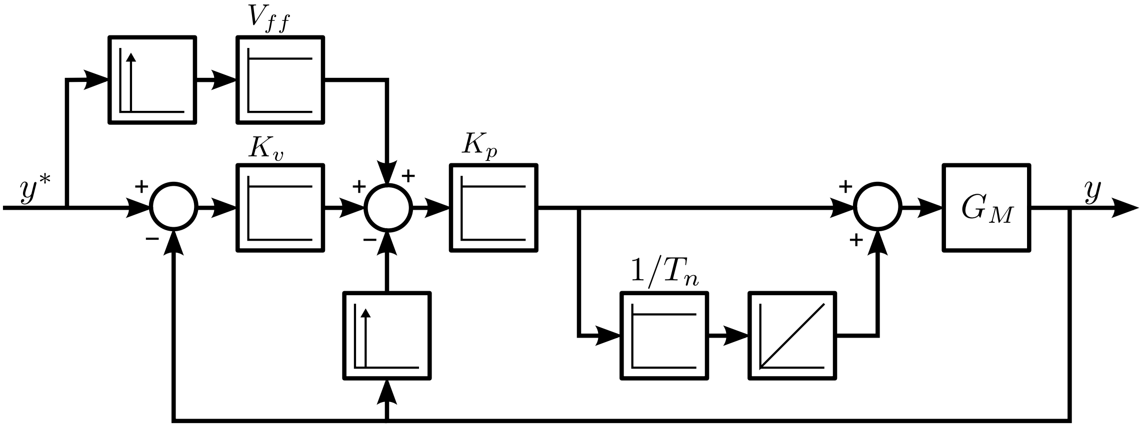
The most common structure for a controller of a feed axis is the cascaded-loop controller as shown in *Fig.* 7.5.

The controller has to be parametrised in an optimal way in order to be able to objectively predict the bandwidth and the dynamic performance of an axis. One method for the parametrisation of a velocity control loop is the method of optimal damping, introduced by Zirn [119]. This method aims to find a velocity gain so, that the resonance at ω_P is damped optimally. The velocity loop gain is denoted as κ and the resulting velocity loop gain for the controller with the plant from *Eq.* (7.15) is

$$\kappa = \frac{K_p}{m}\,. \tag{7.20}$$

The unit of κ is 1/s and the value corresponds to the zero crossing frequency ($0\,\mathrm{dB} = 10^0 = 1$) of the logarithmic plot of the magnitude in the velocity open-loop Bode diagram, if the straight line with a gradient of -20 dB/decade at low frequencies is prolonged virtually (see $\kappa/i\,\omega$ in *Fig.* 7.6).

For a cascaded loop controller without integral part ($T_n \rightarrow \infty$) and velocity feed forward ($V_{ff} = 0$),



K_v	$1/\mathrm{s}$	position controller gain
K_p	kg/s	velocity controller gain
T_n	s	velocity controller integral time
V_{ff}	-	velocity feed forward gain
G_M	s^2/kg	plant transfer function (*Eq.* (7.15))
y^*	m	position set point
y	m	actual position

Figure 7.5: Structure and parameters of a cascaded loop controller

Zirn [119] specified the optimal velocity loop gain as

$$\kappa_{opt} = \frac{1}{2}\,\omega_M \left(1 + \frac{\omega_M^2}{\omega_P^2}\right) \ . \tag{7.21}$$

This parametrisation method delivers not necessarily the minimal worst case error as shown later in *Section* 7.2, but it leads to a conservative first estimation of the resulting controller bandwidth.

For the position controller, a good estimation of the gain K_v is required. As described by Spescha et al. [103], a resulting position controller damping of

$$D_c = \frac{1}{\sqrt{2}} \tag{7.22}$$

is optimal. For the case that $\omega_M \leq \omega_T$, i.e. the antiresonance at the TCP has a higher frequency as the one at the encoder, the calculation

$$K_v = \frac{\kappa}{2} \tag{7.23}$$

as presented by Spescha et al. [103] holds. For the case that $\omega_M > \omega_T$, the value of K_v has to be increased in order to obtain the correct position controller damping.

$$K_v = \frac{\kappa}{2}\,\frac{\omega_M^2}{\omega_T^2} \tag{7.24}$$

has been determined experimentally and is considered a good starting value for that case. For the case that $\omega_M < \omega_T/\sqrt{2}$, no value can be found for K_v, such that the desired damping can be achieved without additional filters.

In summary, for the cases 1A) and 1B) of *Tab.* 7.1 and *Tab.* 7.2, the position controller gain can be calculated with

$$K_v = \frac{\kappa}{2} \cdot \max\left(\left\{1, \frac{\omega_M^2}{\omega_T^2}\right\}\right) \ , \tag{7.25}$$

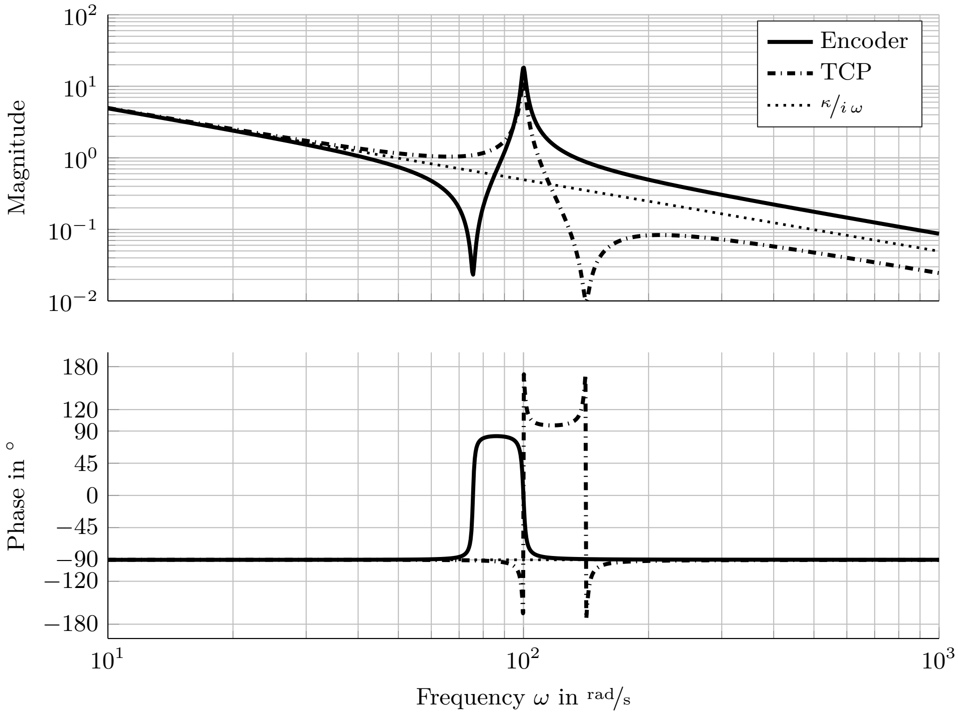


Figure 7.6: Damping-optimal velocity open-loop Bode diagram

if $\omega_M \geq \omega_T/\sqrt{2}$. For the case 1C), no accurate estimation for the correct K_v could be found so far.

The resulting controller bandwidth ω_c can be calculated in good approximation with

$$\omega_c = \frac{\kappa}{\sqrt{2}} = \frac{1}{2\sqrt{2}}\,\omega_M \left(1 + \frac{\omega_M^2}{\omega_P^2}\right) \ . \tag{7.26}$$

For the case 1C), the resulting controller bandwidth is a little higher and, therefore, *Eq.* (7.26) is a conservative estimation. The final controller may not be parametrised with the presented concept but, nevertheless, *Eq.* (7.26) has proven to be a good estimation anyway.

7.1.4.3 Determination of the Critical Eigenfrequency

The requirement on the critical frequencies ω_P and ω_M can be determined with the controller bandwidth estimation from *Eq.* (7.26) and the requirements on the controller bandwidth known from *Eq.* (7.6) or *Eq.* (7.7), for acceleration- or jerk-limited trajectories, respectively. These equations build, together with the resonance and antiresonance frequencies from *Eq.* (7.17), *Eq.* (7.18), and *Eq.* (7.19), constraints on the mechanical and geometrical properties of the structure, which can be considered for an initial design or for parameter modification for a redesign.

7.2 Verification of the Design

In *Section* 7.1.4, the required resonance and antiresonance frequencies of the system have been specified. If these requirements are fulfilled, the required controller bandwidth from *Section* 7.1.3 as well as the required dynamics from *Section* 7.1.2 should be achievable, however, the parametrisation of the controllers could still lead to problems, e.g. if the optimal gain cannot be reached due to stability issues caused by higher-frequency modes. Moreover, if eigenfrequencies arise within the controller bandwidth, e.g. oscillation of the complete machine due to displacement at the support elements, it is not a priori clear, if the disturbance due to these oscillations is acceptable or not. Therefore, it is necessary to verify the design with a full-featured model. Subsequently, the error transfer function from set-point to TCP is calculated and afterwards, a weighting function for the worst-case set-point signal is specified, what enables the calculation of the maximum possible error at the TCP without the need for time-consuming transient simulation.

7.2.1 Dynamic Tracking Error Transfer Function

In order to quantify the trajectory-following accuracy, the dynamic tracking error as defined by Spescha et al. [103] and restated in *Definition* 7.2.1 is used. As mentioned in *Section* 7.1.1, this is the only part of the tracking error, which potentially leads to contour errors, if the axes are synchronised.

Definition 7.2.1. *Let τ be the time delay required such that*

$$\lim_{t\to\infty}\left(y^*(t-\tau)-y(t)\right) = 0 \tag{7.27}$$

for a set point with constant velocity v, i.e. $y^*(t) = v\,t$. Then

$$d(t) := y^*(t-\tau)-y(t) \tag{7.28}$$

is called the dynamic tracking error of the trajectory $y(t)$.

The value of τ depends on control parameters and the control strategy. For a cascaded loop controller according to *Fig.* 7.5 without velocity feed forward ($V_{ff} = 0$), the lag results to

$$\tau = \frac{1}{K_v} \tag{7.29}$$

and for a cascaded loop controller with velocity feed forward ($V_{ff} = 1$), the lag vanishes, i.e.

$$\tau = 0\ . \tag{7.30}$$

The property of interest is the error transfer function $\boldsymbol{E}_{i,i}$ from the set-point of axis i, i.e. y_i^*, to the dynamic tracking error at the TCP in direction i, i.e. d_i. Using the transfer function of a time delay $e^{-s\,\tau_i}$, $\boldsymbol{E}_{i,i}$ can be written as

$$\boldsymbol{E}_{i,i}(s) = \frac{d_i(s)}{y_i^*(s)} = \frac{e^{-s\,\tau_i}\,y_i^*(s)-y_i(s)}{y_i^*(s)}\ . \tag{7.31}$$

This transfer function can be evaluated using a MORe model with Simulink.

7.2.2 Cross-Talk Error Transfer Function

The dynamic tracking error is not the only error occurring when an axis moves. The cross-talk error, i.e. the displacement of the TCP in a direction orthogonal to the axis direction, can reach critical values too. The desired transfer function from the set-point of one axis to the displacement in another direction is zero. The transfer function between the set-point of axis j and the actual value in direction of axis i can be written as

$$\boldsymbol{E}_{i,j}(s) = \frac{y_i(s)}{y_j^*(s)} \qquad \text{for } i \neq j \tag{7.32}$$

and represents the cross-talk error transfer function.

7.2.3 Error Transfer Matrix

For a system with n_a axes, the dynamic tracking error transfer functions and the cross-talk error transfer functions can be combined to a transfer matrix $\boldsymbol{E}$ of dimension $n_a \times n_a$ with the elements

$$\boldsymbol{E}_{i,j}(s) = \begin{cases} \frac{e^{-s\,\tau_i}\, y_j^*(s) - y_i(s)}{y_j^*(s)} & \text{for } i = j \\ \frac{y_i(s)}{y_j^*(s)} & \text{for } i \neq j \end{cases} \quad . \qquad i, j = 1, 2, \ldots, n_a \tag{7.33}$$

7.2.4 Worst-Case Error

The worst-case set-point is a harmonic excitation at the frequency with the maximum error at the TCP, with maximum magnitude without violating any constraints of the axis. The constraints are

- $\hat{y}$, the maximum position amplitude,
- $\hat{v}$, the maximum velocity,
- $\hat{a}$, the maximum acceleration, and
- $\hat{j}$, the maximum jerk.

The harmonic position set-point for limited position, velocity, acceleration, and jerk is

$$y^*(t) = \hat{y}\ \sin(\omega\, t) \tag{7.34}$$

$$y^*(t) = \int v^*(t)\, dt = -\frac{\hat{v}}{\omega}\ \cos(\omega\, t) \tag{7.35}$$

$$y^*(t) = \iint a^*(t)\, dt\, dt = -\frac{\hat{a}}{\omega^2}\ \sin(\omega\, t) \tag{7.36}$$

$$y^*(t) = \iiint j^*(t)\, dt\, dt\, dt = \frac{\hat{j}}{\omega^3}\ \cos(\omega\, t) \tag{7.37}$$

and thus, the position set-point magnitude should not exceed the maximum harmonic amplitude

$$Y(\omega) = \min\left(\left\{\hat{y}, \frac{\hat{v}}{\omega}, \frac{\hat{a}}{\omega^2}, \frac{\hat{j}}{\omega^3}\right\}\right) \tag{7.38}$$

for any oscillation frequency ω. *Fig.* 7.7 shows the set-point magnitude limits for a set of example dynamic and kinematic limits.

The maximum in- and cross-talk errors can be calculated using the worst-case set-point Y from *Eq.* (7.38) and the error transfer matrix $\boldsymbol{E}$ from *Eq.* (7.33) with

$$\hat{\boldsymbol{E}} = \max_{\omega}\left(|\boldsymbol{E}(i\omega)\, Y(i\omega)|\right) \ . \tag{7.39}$$

For the special case of unlimited jerk, a position controller with damping $D \approx {}^1/\sqrt{2}$, and the worst-case trajectory from *Fig.* 7.2(a), the maximum error exceeds the value calculated in *Eq.* (7.39) by a factor of approximately 1.35. This factor is investigated experimentally. In order to calculate the worst case error for acceleration-limited trajectories, therefore, the limit from *Eq.* (7.38) has to be corrected to

$$Y_a(\omega) = \min\left(\left\{\hat{y}, \frac{\hat{v}}{\omega}, 1.35\,\frac{\hat{a}}{\omega^2}\right\}\right) \ . \tag{7.40}$$

For jerk-limited trajectories, *Eq.* (7.38) has to be corrected by a factor of 1.1 in order to accord with the worst-case trajectory *Fig.* 7.2(b) and thus, the worst case error for jerk-limited trajectories is

$$Y_j(\omega) = \min\left(\left\{\hat{y}, \frac{\hat{v}}{\omega}, \frac{\hat{a}}{\omega^2}, 1.1\,\frac{\hat{j}}{\omega^3}\right\}\right) \ . \tag{7.41}$$

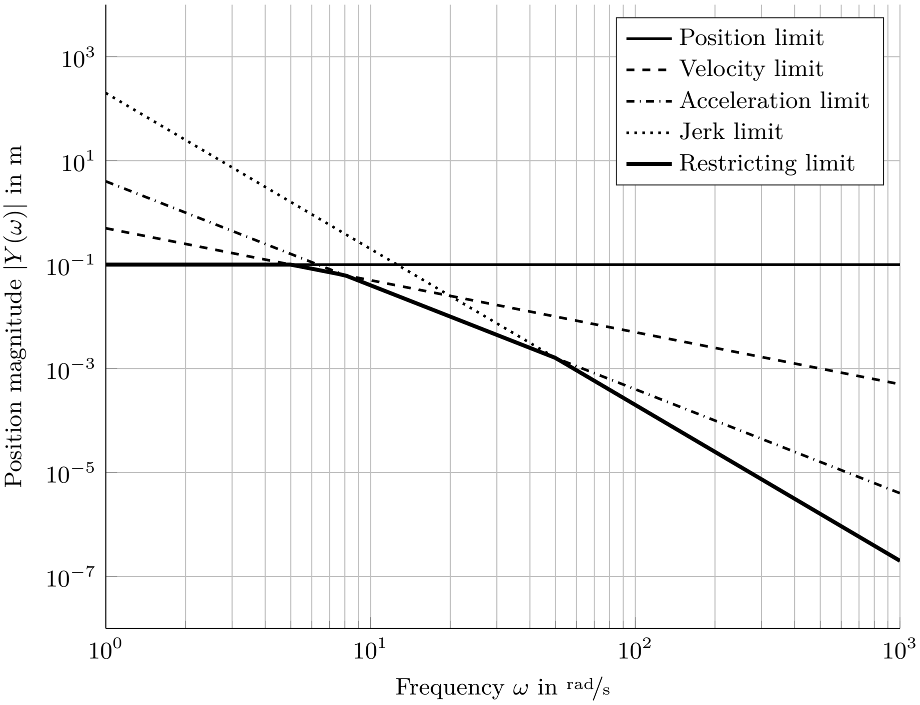


Figure 7.7: Example for position set-point magnitude $Y(\omega)$ with dynamic and kinematic limits according to Equations 7.34 to 7.38 ($\hat{y} = 0.1$ m, $\hat{v} = 0.5$ m/s, $\hat{a} = 4$ m/s², $\hat{j} = 200$ m/s³)

7.3 Discussion of the DtS Approach

The key idea of DtS is to enable design engineers to work goal-oriented on a design that

- fulfils the specifications on accuracy and dynamics and
- is not oversized.

Using the specifications on the critical eigenfrequency and the antiresonance, and with the help of geometrical relations between the locations of force application, measurement system, and TCP, the design engineer is able to evaluate fundamental dimensions like linear guide distance, position of the centre of mass, position of measurement systems, and stiffness of machine components.

As soon as the first component of the machine tool is designed, the model process can be started so that the complete model of the machine tool can be finalised very soon after the design is completed. This model can then be utilised for an efficient analysis of the actual dynamic performance of the system by means of worst-case error FRFs. Detailed analysis of the oscillation shapes at the bottleneck, i.e. the frequencies with the largest error magnitude, helps for an effective redesign of the structure, if the specifications are not yet fulfilled. At this stage, it is advisable to consider known model uncertainties and to specify a safety margin for the specifications.

After having found a design which fulfils the specifications in the simulative analysis, the machine can be manufactured and analysed by means of measurements. If no errors occurred during modelling, the specifications should also be fulfilled with the actual machine. If not, the model has to be corrected and a new redesign-iteration has to be started.

In short, the design tasks are:

1. Determine the dynamic limits needed to achieve the desired productivity using CNC trajectory planning functionality (*Section* 7.1.2).
2. Determine the required resonance and antiresonance frequency (*Section* 7.1.4).
3. Design the structure. Use the requirements as a guideline to find design parameters as stiffness, masses, and dimensions for the new structure.

4. Model the structure with the software MORe (*Chapter* 6).
5. Calculate the maximum error at the TCP (*Section* 7.2).
6. Iterate the design tasks until the specifications are fulfilled in simulation.
7. Manufacture and analyse the system by means of measurements. Iterate the design if needed.

Irrespective of whether the manufactured machine reaches the specified dynamics and accuracy or not, the measurement data should be used to validate and, if required, correct the model. This expands the database of validated component models and increases the reliability of future models for similar machines.

This design process considers the dynamic accuracy only. However, other analyses like static deformation due to gravity, backlash due to static friction or mechanical play, or process stability are also possible by means of the model.

8 Application

In this chapter, the framework for simulation and analysis of the dynamics of machine tools presented in the preceding chapters is applied to a test bench. First, the test bench is briefly described and the corresponding MORe model is introduced, then, in *Section* 8.2, the model accuracy is determined by means of an experimental modal analysis, Bode diagrams, and contour error plots. In *Section* 8.3, the estimations underlying the Design to Specifications process are evaluated. In *Section* 8.4, an application of a MORe model for the analysis of position dependencies is presented and the results of this chapter are discussed in *Section* 8.5.

8.1 Andromeda Gantry Stage

The test bench used here is a two-axis positioning system with a gantry axis X, consisting of the axes X1 and X2, and a second axis Y in Cartesian arrangement. All axes are realised with direct drives and linear encoders. The test bench is named Andromeda and pictured in *Fig.* 8.1.

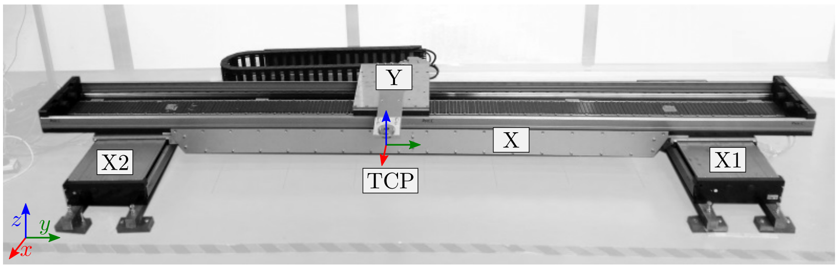
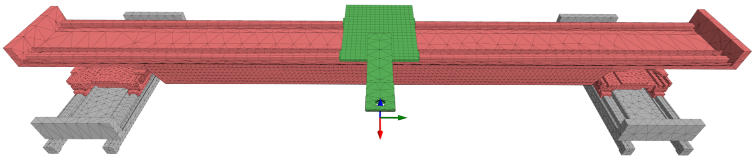


Figure 8.1: Test bench Andromeda gantry stage

The Andromeda stage has originally been used by Garcia-Herreros Landazabal [45]. Ryser [87] redesigned the stage, especially the joints coupling the bridge to the X drives.

The structure of the MORe model is shown in *Fig.* 8.2. The simplified CAD model and main parts of the finite element model have been established by Ryser [87] and the major work on the corresponding MORe model has been accomplished by Ceresa [22] in the context of a bachelor's and master's thesis, respectively.

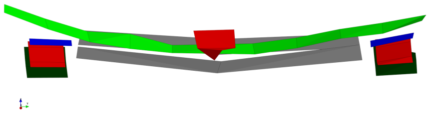
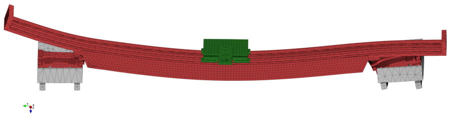
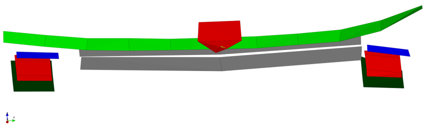
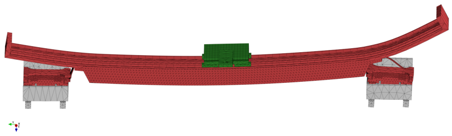
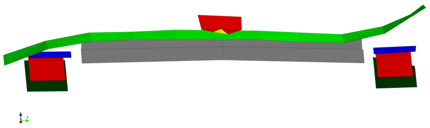
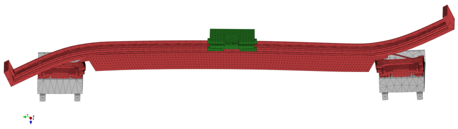
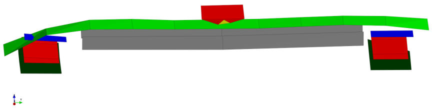
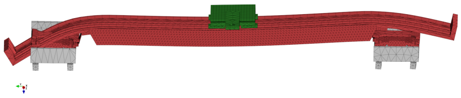
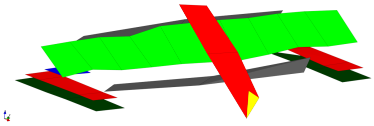
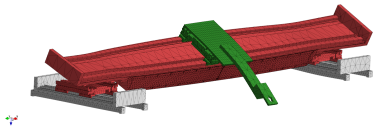
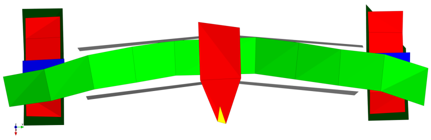
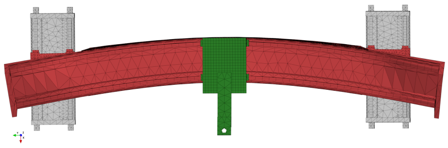
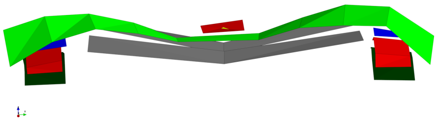
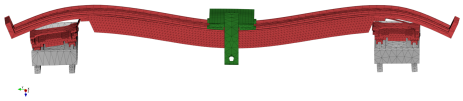


	Component	N	n
⬛	Basis	98 598	224
🟥	X-Axis	292 869	412
🟩	Y-Axis	42 510	49
	Total	433 977	685

Figure 8.2: MORe model of the Andromeda stage. The table on the right indicates the individual components with the order of the systems before reduction, N, and after reduction, n.

The composition consists of four components, two base parts, the X-axis, and the Y-axis. The bases feature each four support elements as stationary interfaces and two linear guide rails, modelled as

Table 8.1: Comparison of mode shapes and eigenfrequencies between experimental and simulative modal analysis for the axes positioned at $x = 0$ and $y = 0$

Mode No. Rel. error	**Experimental Modal Analysis** Shape Frequency	**Simulative Modal Analysis** Shape Frequency
1		
-2.2%	231 rad/s	226 rad/s
2		
-5.3%	263 rad/s	249 rad/s
3		
-5.2%	307 rad/s	291 rad/s
4		
-6.4%	329 rad/s	308 rad/s
5		
1.4%	368 rad/s	373 rad/s
6		
-9.1%	828 rad/s	753 rad/s
7		
-5.3%	828 rad/s	784 rad/s

moving interfaces with five harmonics. Furthermore, the basis components have moving interfaces for the secondary part of the linear motor and the linear scale, also modelled with five harmonics each.

The X-axis has eight stationary interfaces for the linear guide carriages of the axes X1 and X2, and two moving interfaces for the linear guides. The length of the linear guides for the Y-axis is 3.1 m and the moving interfaces are modelled with 20 harmonics, leading to a force application width of ca. 150 mm, what is slightly more than the length of the carriages of 120 mm, but is considered as sufficient to model the locality of the force application. The force application to the permanent magnets of the motor for the Y-axis and the linear scale have equally been modelled using moving interfaces with 20 harmonics. Furthermore, the primary parts of the motors and the encoder heads for the X1- and X2-axes are modelled as stationary interfaces.

The Y-axis has four linear guide carriages, a motor, an encoder head and the TCP modelled as stationary interfaces.

All interfaces are defined according to the methods presented in *Chapter* 4 and the components are reduced by means of the KMS-based model order reduction method introduced in *Chapter* 3. However, because each interface degree of freedom leads to an additional order of the reduced system, only those degrees of freedom which are required for simulation are considered for model order reduction. For the linear guides, these are all translational degrees of freedom and the rotational degrees of freedom are neglected, because the rotational stiffness of an axis is usually dominated by pairs of carriages and their translational stiffness. The motors and linear scales are modelled with the axial degree of freedom only and for the TCP, all six degrees of freedom are retained.

The reduction has been parametrised for a frequency range of 3000 rad/s with an accuracy limit of 5%. The total number of degrees of freedom is 433 977 for the original finite element models and 685 for the reduced components. For all components equal Rayleigh damping values of $\alpha = 10$ and $\beta = 10^{-5}$ are used. These values are a rough estimation of the structural damping and the exact values are of minor importance. However, for transient simulation efficiency reasons, it is important to use values different from zero.

The support elements and the linear guide couplings are modelled as links with stiffness link properties according to *Section* 5.3.1. For the linear guides, the stiffness values provided by the supplier are used. Static friction is modelled in Simulink with 15.5 N per carriage, as measured by Ryser [87]. No viscous damping is modelled, because it is regarded of minor importance in this case. Further links are defined for the motors and linear measurement systems in order to be used as inputs and outputs for the controllers.

The controllers are realised as proportional cascaded-loop controllers without any non-linearities, feed-forward, or integrating components. The controller gains are approximately set-up for optimal damping according to *Section* 7.1.4.2.

8.2 Model Accuracy

For the evaluation of the model accuracy, results from an experimental modal analysis, Bode diagrams, and a test contour are used.

The results of the experimental modal analysis from Ceresa [22] are used. For the modal analysis, the structure has been excited using an impact hammer and the oscillations at a multitude of points on the structure have been measured by means of piezoelectric accelerometers. From the resulting FRFs, mode shapes and eigenfrequencies have been fitted using the software solution ME'scope [73]. The identified modes are shown in *Tab.* 8.1 side by side with the modes evaluated using the MORe model. The mode shapes match excellently and the relative errors of the eigenfrequencies are below 10%.

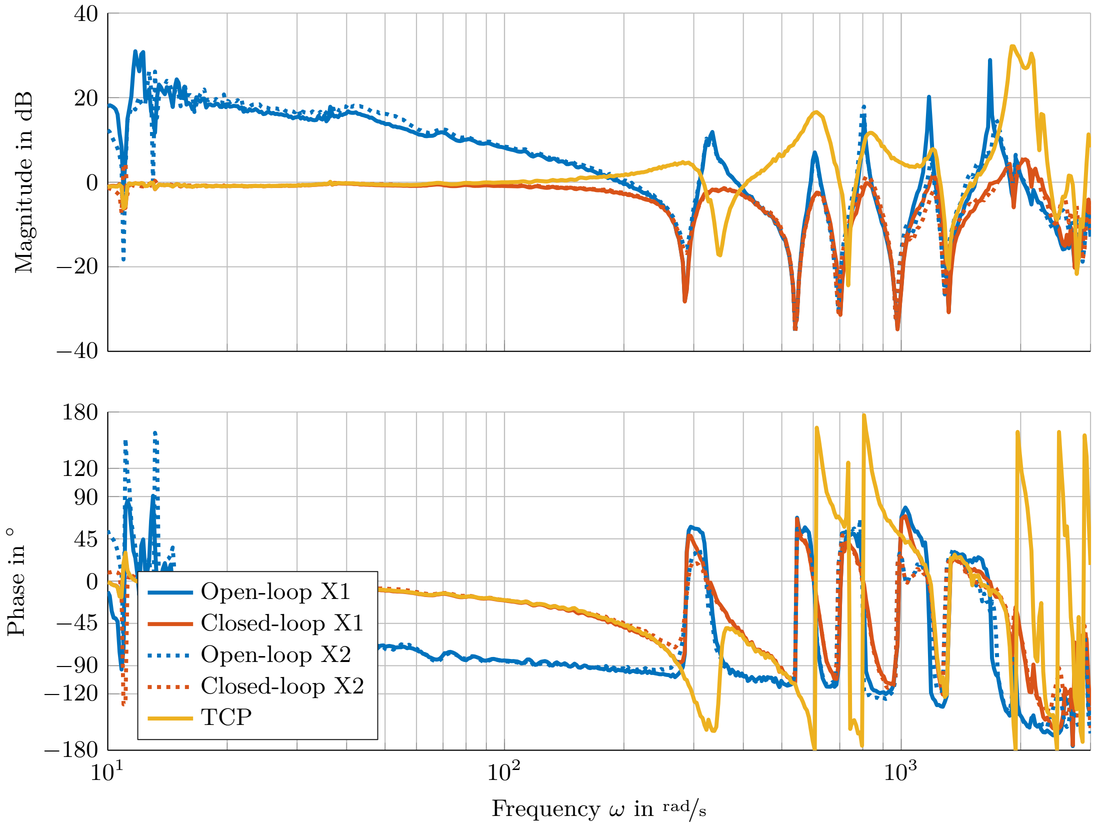
Frequency responses have been measured by means of sine sweep signals with constant velocity magnitude and a velocity offset used to avoid zero crossings of the velocity, what leads to inaccurate results due to static friction. The data for the TCP has been measured by means of a Heidenhain KGM grid encoder. *Fig.* 8.3 and 8.4 show measured and simulated Bode diagrams for the velocity loop of the X- and Y-axis, respectively. Both open- and closed-loop FRFs as well as the resulting FRF at the TCP for the closed loop are shown. Similarly, in *Fig.* 8.5 and 8.6, the position loop Bode diagrams for both axes are figured. The first figure in each case shows the measured results, whereas the second figure shows the simulated counterpart.

Overall, the results are of very good accordance. The most important characteristics are the first antiresonance and resonance. Their frequencies, and especially the order of them both should match accurately in order to reach a good estimation of the performance of the machine. Moreover, the following antiresonances and resonances should match qualitatively in order to be able to parametrise the controller correctly. If the order of an antiresonance and resonance pair is switched, the phase changes its characteristics and the controller parametrisation has to be different.

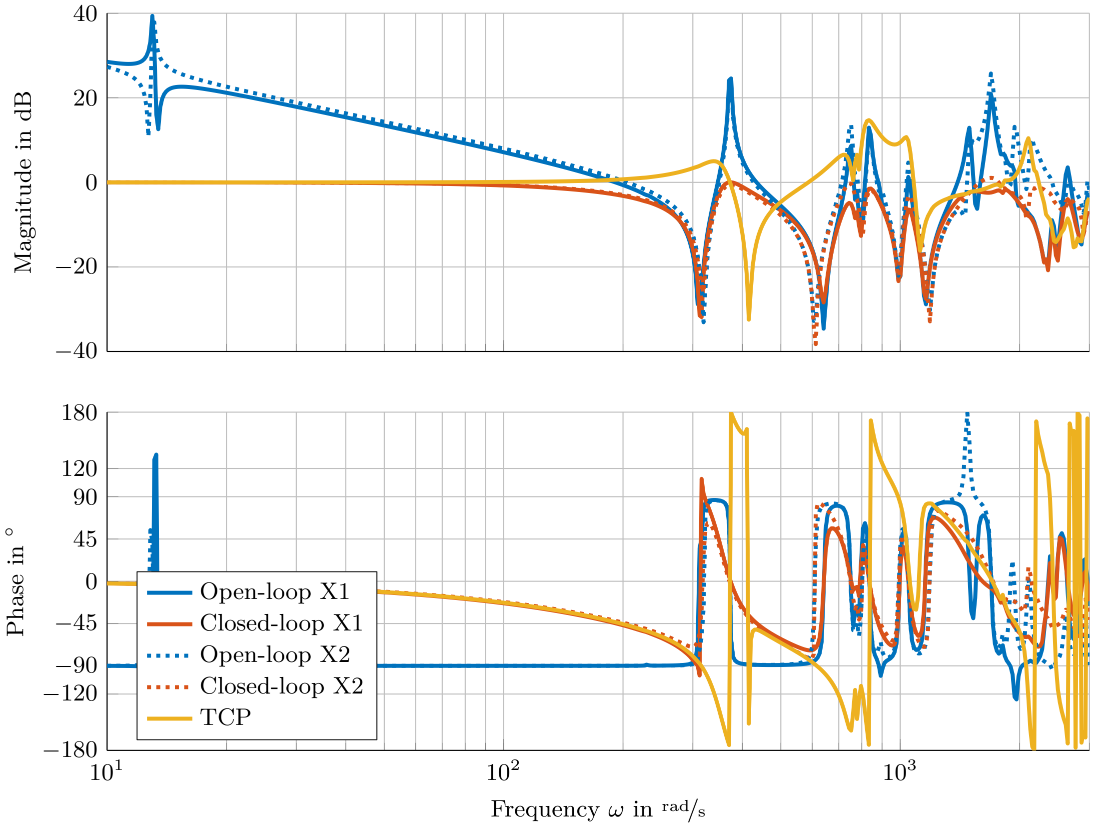
The velocity loop results for the X-axes in *Fig.* 8.3 show a decent quality. With exception of a difference in frequency of the first resonance of about 11%, the qualitative behaviour is of good accordance. Similarly, for the subsequent antiresonance-resonance pairs, the qualitative behaviour matches, but the frequency is slightly shifted up or down. Similarly, the results for the Y-axis also show a good accordance. The measured Bode diagrams for the Y-axis show significantly more noise than the ones of the X-axis. However, the qualitative behaviour is apparent. As a consequence of the good quality of the velocity loop transfer functions, the results for the position loop do also match satisfyingly.

In *Fig.* 8.7, the contour errors for a test geometry are shown. The errors are scaled by a factor of 50 orthogonally to the contour. The test geometry is built of half- and quarter-circle sections concatenated to a rounded cross. This contour is near the worst case for a cascaded-loop controlled lumped mass as shown in *Fig.* 7.2(a), because at the points where the curvature changes, for one axis an acceleration step occurs from positive maximum to negative maximum acceleration and vice versa, whilst the other axis travels with near-constant velocity and the dynamic tracking error of the first axis does completely result as contour error. The comparison between measurement and simulation shows a decent accordance, both qualitatively and quantitatively. Especially the discrepancy between TCP and encoder of the simulation resembles the measured one. The maximum contour error at the TCP is 105 μm for the measurement and 110 μm for the simulation.

The comparisons between measurement and simulation show that the model quality is adequate for the estimation of the dynamic performance of the Andromeda stage. It is pointed out here that the model parameters of the structure have not been fitted in any way in order to match the measurements. The simplified geometry, standard material properties, and stiffness values provided by the suppliers have been used.

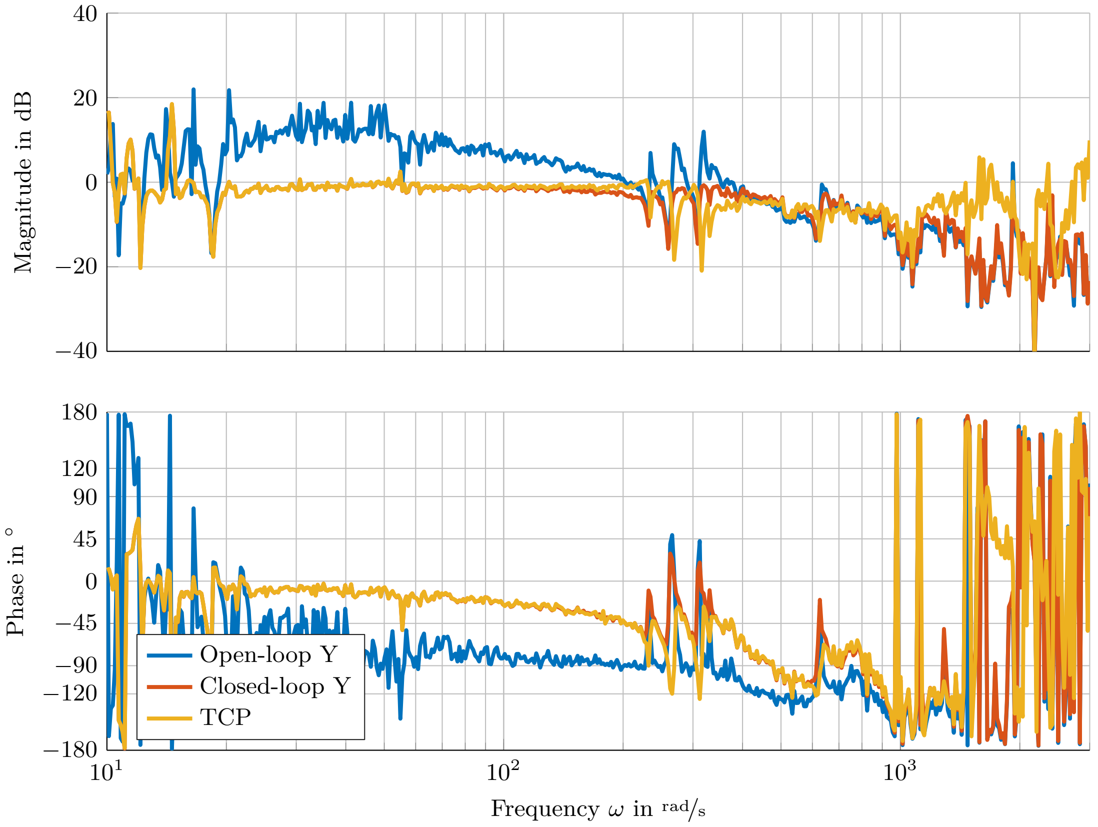


(a) Measurement

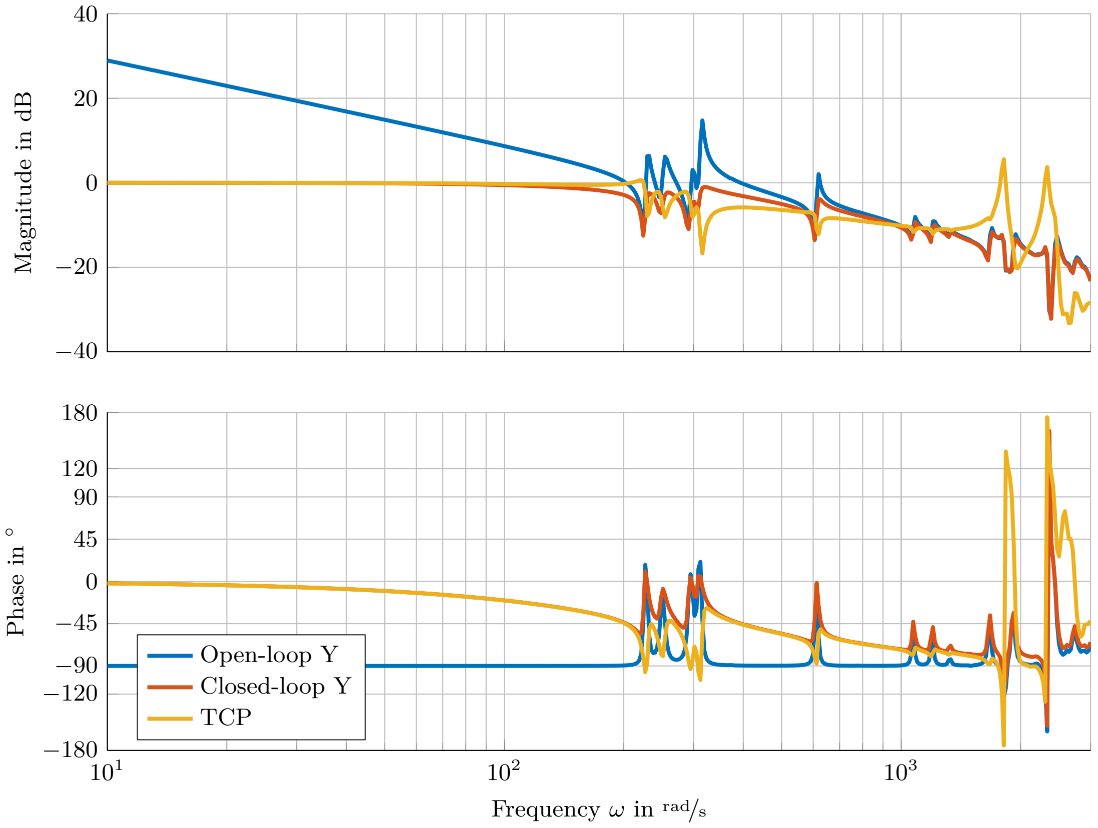


(b) Simulation

Figure 8.3: Velocity loop Bode diagrams for the X-axes for the axes positioned at $x = 0$ and $y = 0$

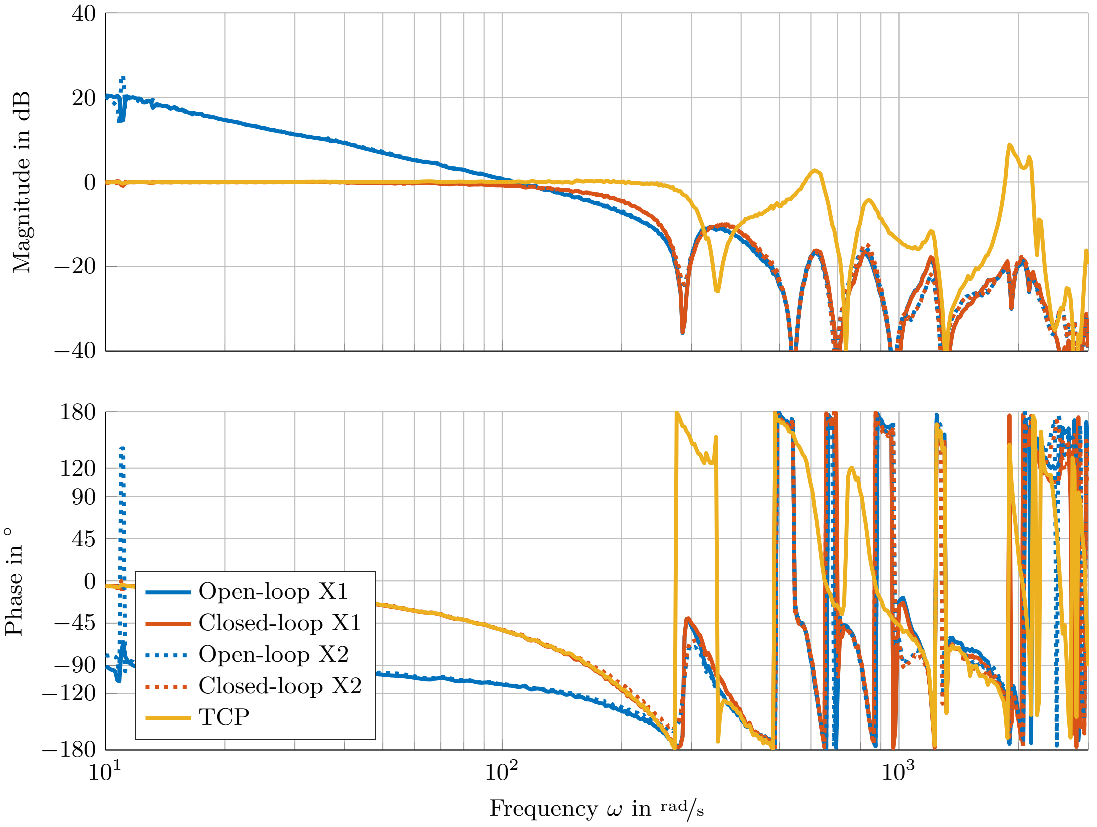


(a) Measurement

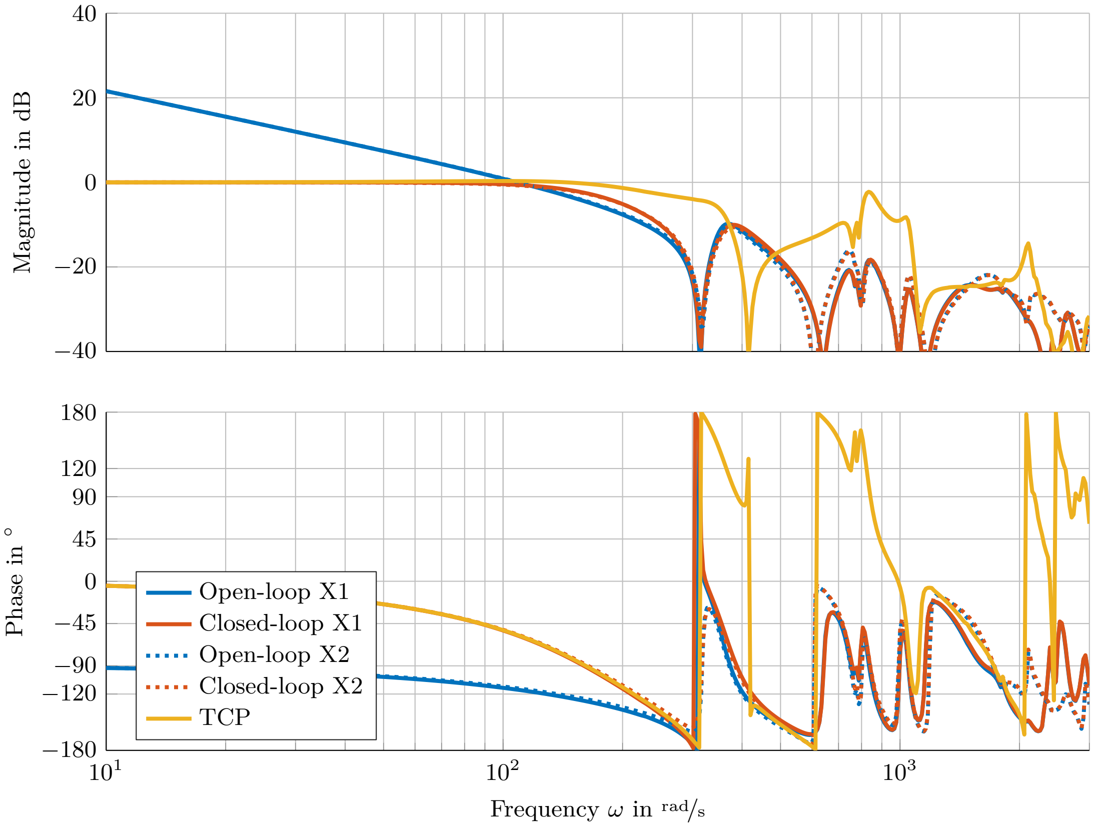


(b) Simulation

Figure 8.4: Velocity loop Bode diagrams for the Y-axis for the axes positioned at $x = 0$ and $y = 0$

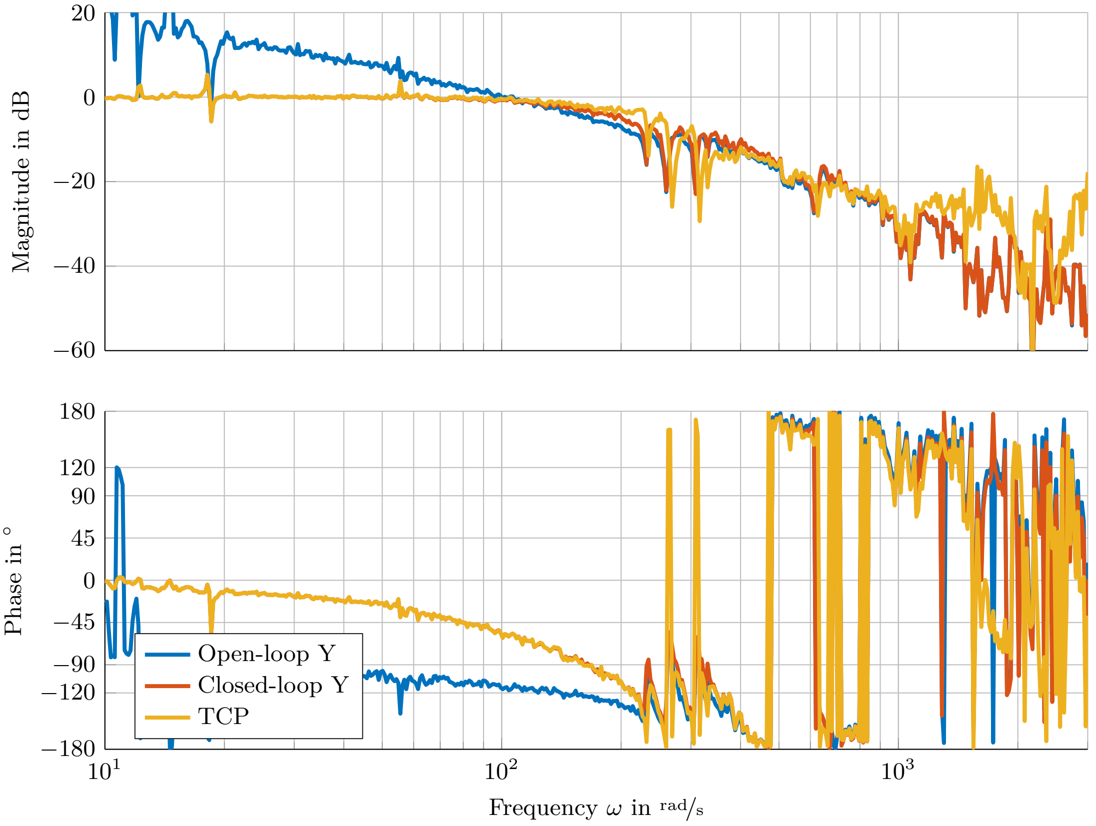


(a) Measurement

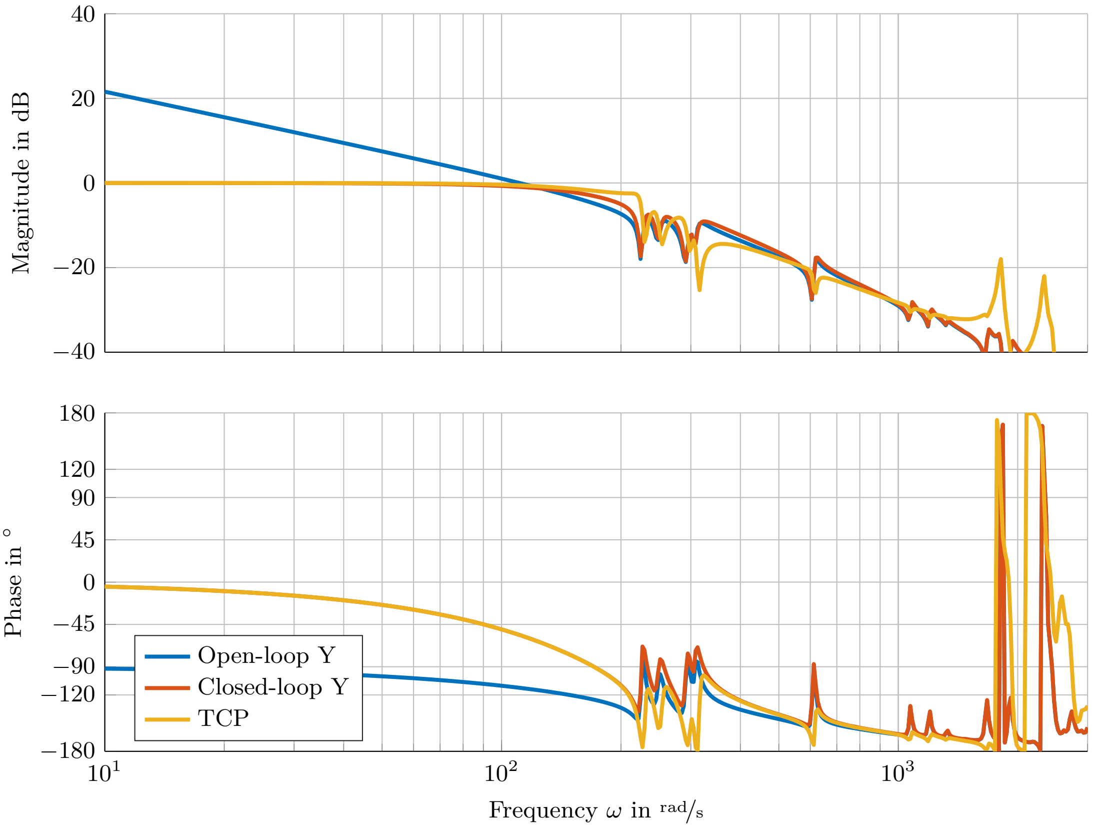


(b) Simulation

Figure 8.5: Position loop Bode diagrams for the X-axes for the axes positioned at $x = 0$ and $y = 0$

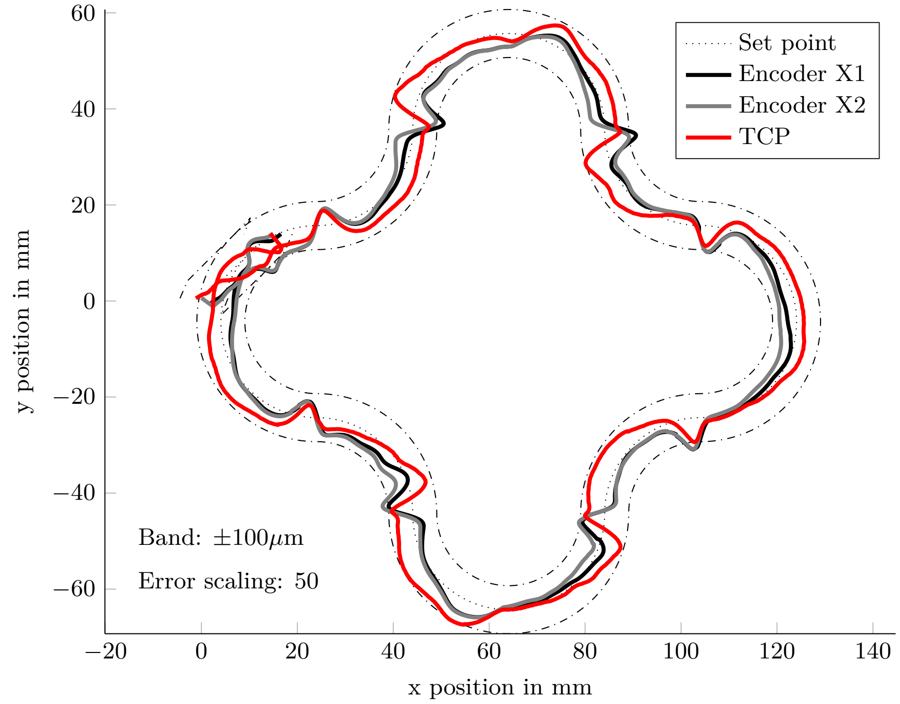


(a) Measurement

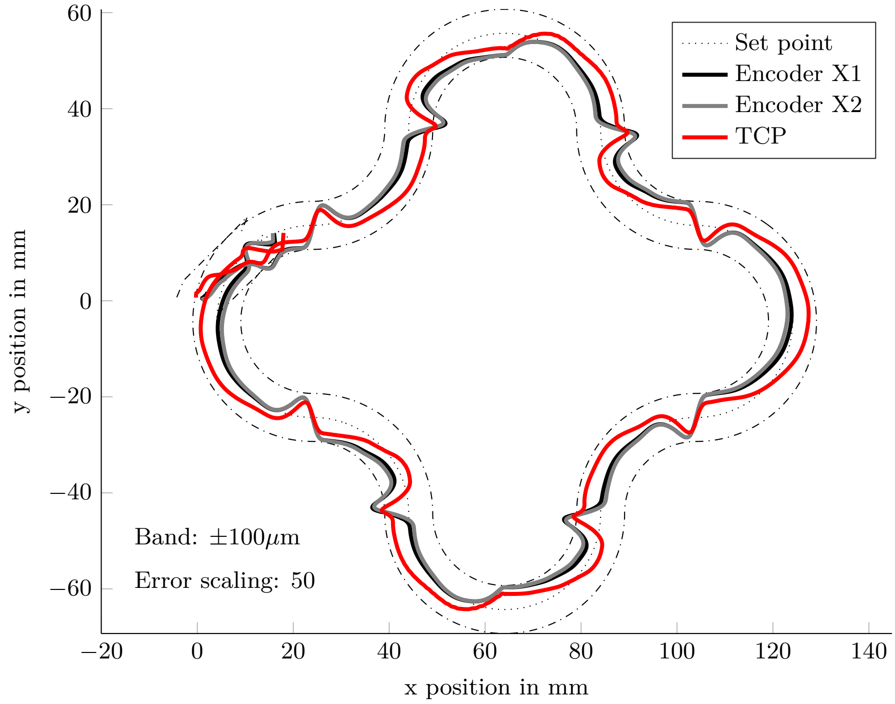


(b) Simulation

Figure 8.6: Position loop Bode diagrams for the Y-axis for the axes positioned at $x = 0$ and $y = 0$



(a) Measurement



(b) Simulation

Figure 8.7: Contour errors at the encoders and TCP for rounded cross trajectories with maximum acceleration $\hat{a} = 9\,\mathrm{m/s^2}$ for a starting position of $x = 0$ and $y = 0$

8.3 Design to Specifications Evaluation

In *Chapter* 7, the Design to Specifications procedure has been introduced. There, the derivation of requirements on resonances and antiresonances is based on a heavily simplified system of forth order. Furthermore, a method for verification of a specific design by means of simulation has been introduced. Here, both the derivation of requirements and the performance analysis are evaluated by means of the Andromeda model.

8.3.1 Evaluation of the Actual Error

The actual contour error for the rounded cross contour shown in *Fig.* 8.7 is 110 μm. However, the error estimation from *Section* 7.2 is restricted to dynamic errors caused by structural vibrations and, especially, the controller only. Deviations due to non-linear forces like static friction are to be examined separately. However, they are contained in the measured contour error, and therefore, for the verification of error estimations, the contour error of a rounded cross simulated with disabled static friction is used. Furthermore, the position controller has been fine-tuned in order to achieve a damping factor of $1/\sqrt{2}$ for the closed-loop transfer function to the TCP.

The simulated contour is shown in *Fig.* 8.8. Due to the optimal damping factor, the error at the TCP almost vanishes on the circular arcs and reaches the maxima at the curvature changeovers. This allows a precise readout of the maximum errors caused by the X- and Y-axes. For the X-axis, the maximum error reaches 76 μm and the one for the Y-axis is 138 μm.

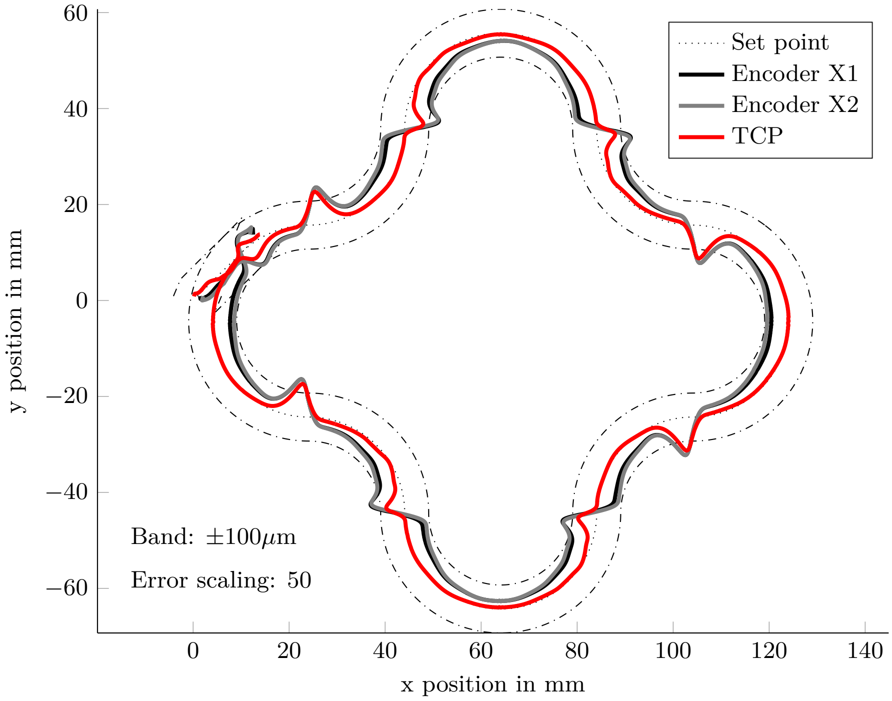


Figure 8.8: Contour errors at the encoders and TCP for a rounded cross contour, simulated without static friction and with fine-tuned position controller damping for a starting position of $x = 0$ and $y = 0$

8.3.2 Error Estimation by Means of Antiresonance and Resonance Frequencies

In *Section* 7.1 the requirements on a structure have been derived from the dynamics and accuracy specifications for the axes. The results are requirements on the resonance and antiresonance frequencies of the axes. Here, these specifications are assessed by applying the calculations in reverse.

The resonance and antiresonance frequencies for both axes are listed in *Tab.* 8.2 together with the resulting velocity loop gain for optimal damping of the structural modes, the estimated position controller bandwidth, the estimated maximum dynamic tracking error, and for comparison, the actual maximum contour error evaluated using the rounded cross contour. The origin for the read-out and derivation of the values is also listed in this table.

For the X-axis, the maximum error is estimated at $102\,\mu\mathrm{m}$ and the actual value is $76\,\mu\mathrm{m}$. The error is thus overestimated by ca. 34%. The maximum error for the Y-axis is estimated at $164\,\mu\mathrm{m}$ and the actual value is $138\,\mu\mathrm{m}$, what is an overestimation of 19%. Against the background that the estimation takes a low number of structural parameters into account, this result is satisfactory. Moreover, it is emphasised here that the rounded cross contour is not the worst-case contour for the present controller transfer function.

Table 8.2: Dynamic tracking error estimation by means of resonance and antiresonance frequencies for a maximum acceleration of $\hat{a} = 9\,\mathrm{m/s^2}$

Description	**Symbol**	**X-Axis**	**Y-Axis**	**Origin**
Resonance	ω_P	$374\,\mathrm{rad/s}$	$230\,\mathrm{rad/s}$	*Fig.* 8.3(*b*) / 8.4(b)
Antiresonance at encoders	ω_M	$320\,\mathrm{rad/s}$	$224\,\mathrm{rad/s}$	*Fig.* 8.3(*b*) / 8.4(b)
Antiresonance at TCP	ω_T	$415\,\mathrm{rad/s}$	$231\,\mathrm{rad/s}$	*Fig.* 8.3(*b*) / 8.4(b)
Velocity loop gain for optimal damping	κ_{opt}	$277\,\mathrm{1/s}$	$218\,\mathrm{rad/s}$	*Eq.* (7.21)
Estimated position controller bandwidth	ω_c	$196\,\mathrm{rad/s}$	$154\,\mathrm{rad/s}$	*Eq.* (7.26)
Estimated max. dynamic tracking error	$\hat{d}_a$	$102\,\mu\mathrm{m}$	$164\,\mu\mathrm{m}$	*Eq.* (7.4)
Actual max. contour error	$\hat{e}_c$	$76\,\mu\mathrm{m}$	$138\,\mu\mathrm{m}$	*Fig.* 8.8

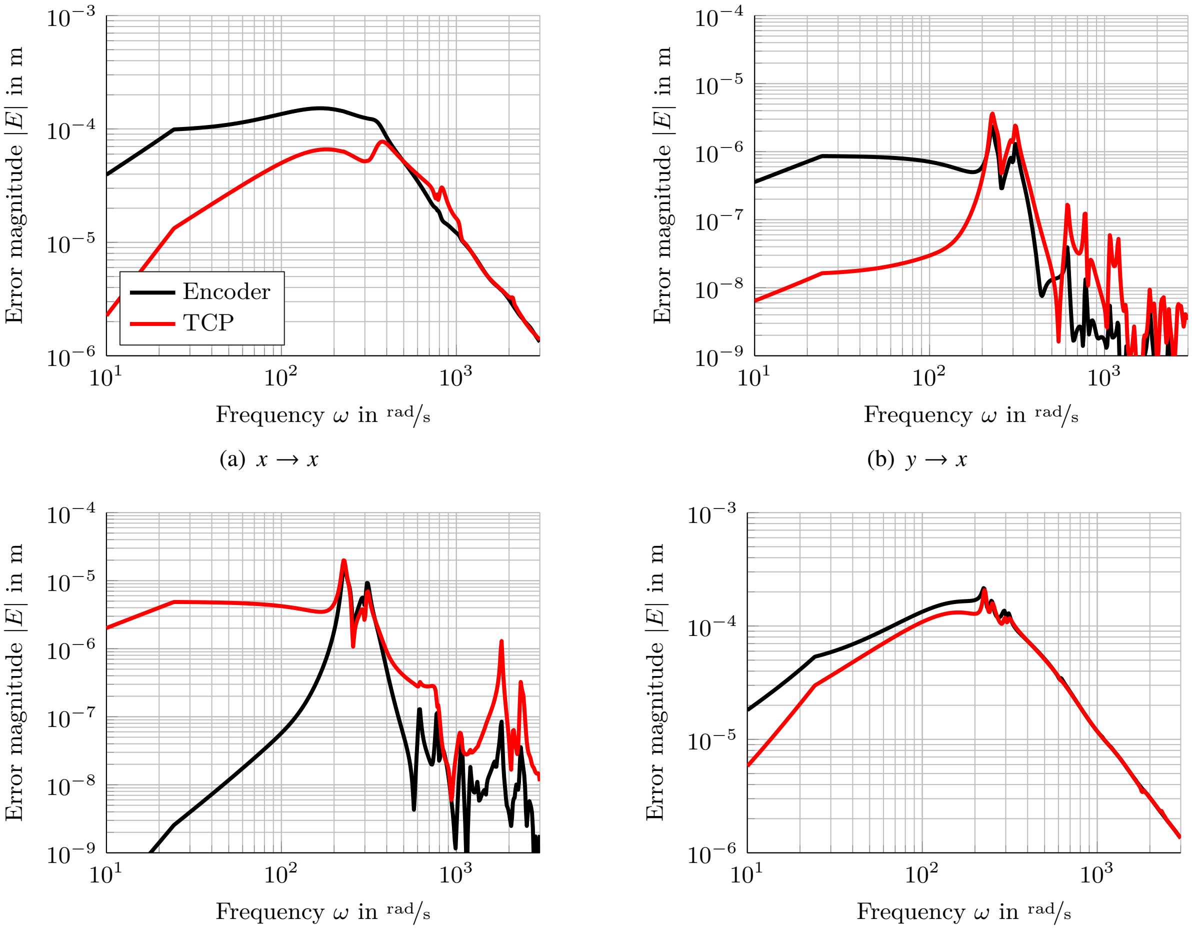
8.3.3 Error Estimation by Means of Error Transfer Functions

The estimation presented in the last section does neither consider the actual controller structure and parametrisation nor the effect of further modes. Thus, a more precise estimation that is based on error transfer functions of the controlled MORe model has been presented in *Section* 7.2. The maximum error transfer matrix $\hat{\boldsymbol{E}}$ from *Eq.* (7.39) realised using the weighting function for acceleration limited and jerk unlimited trajectories from *Eq.* (7.40) are shown for the Andromeda model in *Fig.* 8.9. The four figures correspond to the four elements of the matrix for a system of two axes. On the diagonal, the maximum dynamic tracking errors for both axes are shown, off the diagonal, the cross-talk errors for excitation at one axis and measurement at the other axis are shown.

In this case, the cross-talk errors are about ten times smaller then the dynamic tracking errors. The maximum dynamic tracking error for the X-axis is $77\,\mu\mathrm{m}$ and the one for the Y-axis is $207\,\mu\mathrm{m}$. The dynamic tracking error for the X-axis corresponds nearly exactly to the maximum contour error from *Fig.* 8.8. The one for the Y-axis is 50% higher than the value of $138\,\mu\mathrm{m}$, read out from *Fig.* 8.8. This, however, can be explained by the lowly damped resonance at $227\,\mathrm{rad/s}$ in *Fig.* 8.9(*d*). This value is reached only when this resonance frequency is excited explicitly, e.g. by repeated circles with maximum acceleration at this frequency. This has also been proven by simulation and thus, the error estimation by means of the weighted transfer functions has proven to be very accurate, and most important, delivers the maximum error for the worst case, independently of specific test trajectories.

8.4 Spatial Analyses

Moving interfaces have been introduced in *Section* 4.3 in order to enable analysis of position dependent static and dynamic properties of machine tool axes. The tilting of the Y-axis carriage depending on



(a) $x \longrightarrow x$

(b) $y \longrightarrow x$

(c) $x \longrightarrow y$

(d) $y \longrightarrow y$

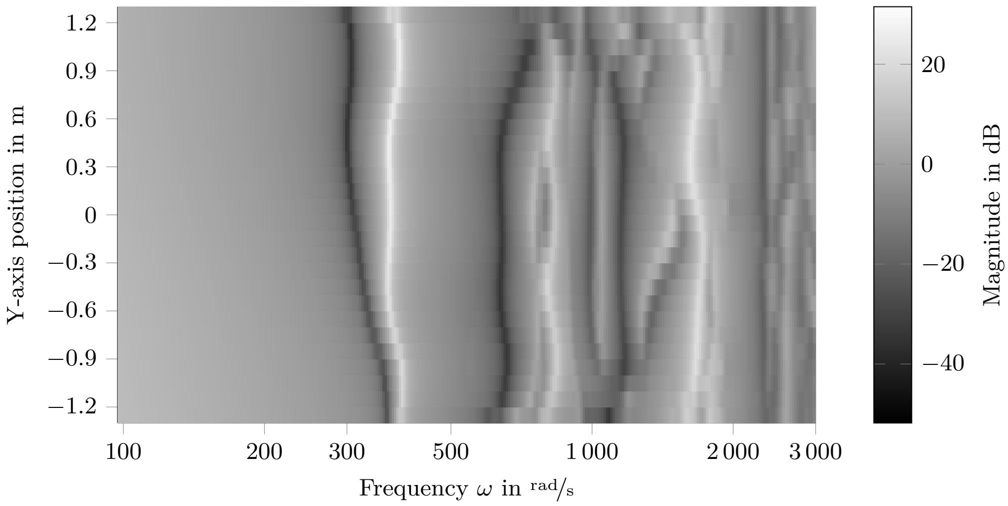
Figure 8.9: Error FRFs according to *Eq.* (7.39) for the Andromeda stage and for trajectories with unlimited jerk, maximum acceleration of $\hat{a} = 9\,\mathrm{m/s^2}$, maximum velocity of $\hat{v} = 0.5\,\mathrm{m/s}$, and maximum travel of $\hat{y} = 0.2\,\mathrm{m}$ for the axes positioned at $x = 0$ and $y = 0$

the Y-axis position has been simulated and successfully verified with measurements by Lanz, Spescha, Ryser, Ceresa, and Weikert [62] by adding different masses near the TCP of the Andromeda stage.

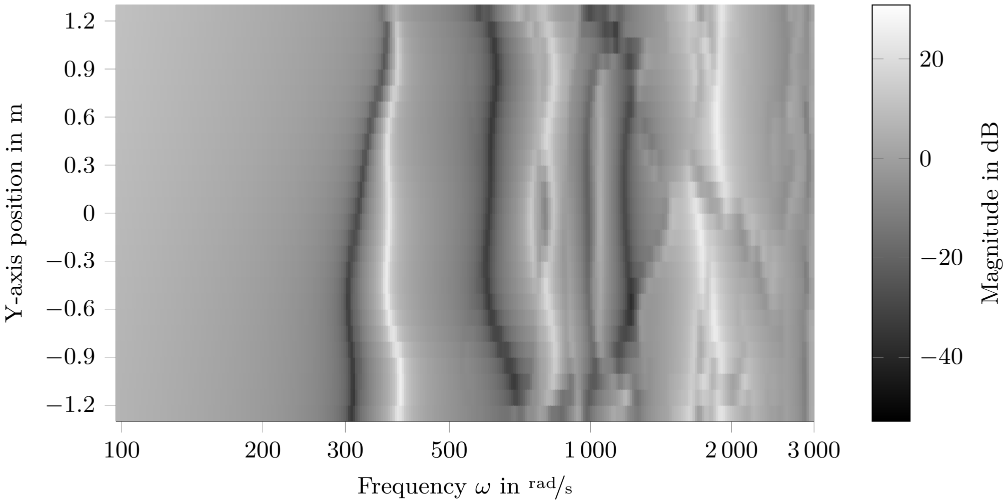
In the following, the dynamic behaviour within the travel range of the Y-axis is analysed. The velocity loop Bode diagrams for the X-axis shown above in *Fig.* 8.3(*b*) have been evaluated for the Y-axis placed in the centre of its travel range, i.e. at position 0. For the analysis of the dependency of the behaviour of the X-axis on the position of the Y-axis, in *Fig.* 8.10, the magnitude of the velocity open-loop Bode diagrams for the X1- and X2-axes are shown, calculated for 27 points along the Y-axis.

The lighter lines are the resonances and the dark lines are the antiresonances. The lowest-frequency antiresonance and resonances are from special interest, because they determine the velocity loop gain for optimal damping. The resonance frequency varies between 370 and 400 rad/s and the antiresonance is even more sensitive and varies between 300 and 370 rad/s. The characteristics of the antiresonances for the X1- and X2-axis are very similar, but mirrored at the position 0. This leads to different dynamics for the controllers of both axes for any position but the centre.

The maximum dynamic tracking error has been evaluated for all positions along the axis without changing the parametrisation of the controller, leading to 27 error transfer matrices according to *Fig.* 8.9. The maximum value for the dynamic tracking error of both axes at each position is combined to the curves shown in *Fig.* 8.11. These curves describe the dynamic performance along the travel of the X-axis. The minimum value can be found near the centre and the maximum value for the X-axis is on



(a) X1-axis



(b) X2-axis

Figure 8.10: Velocity open-loop FRFs depending on the position of the Y-axis

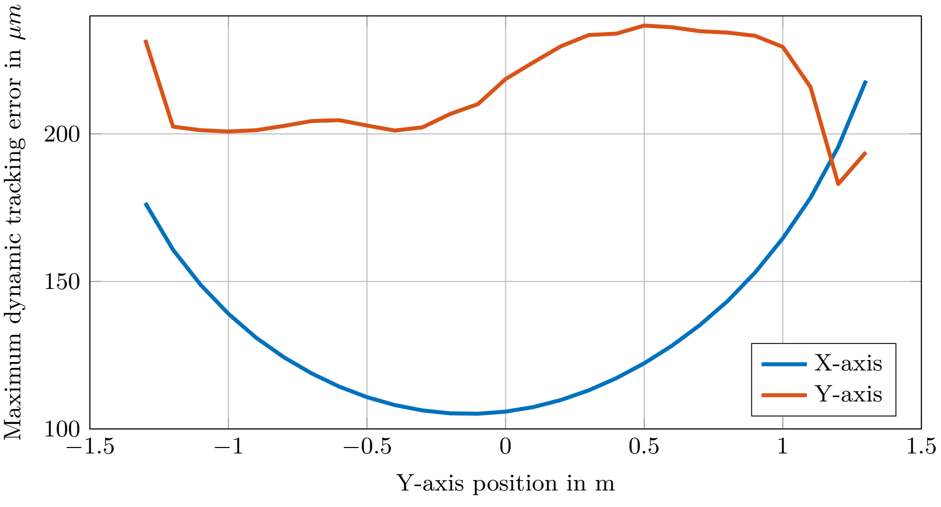
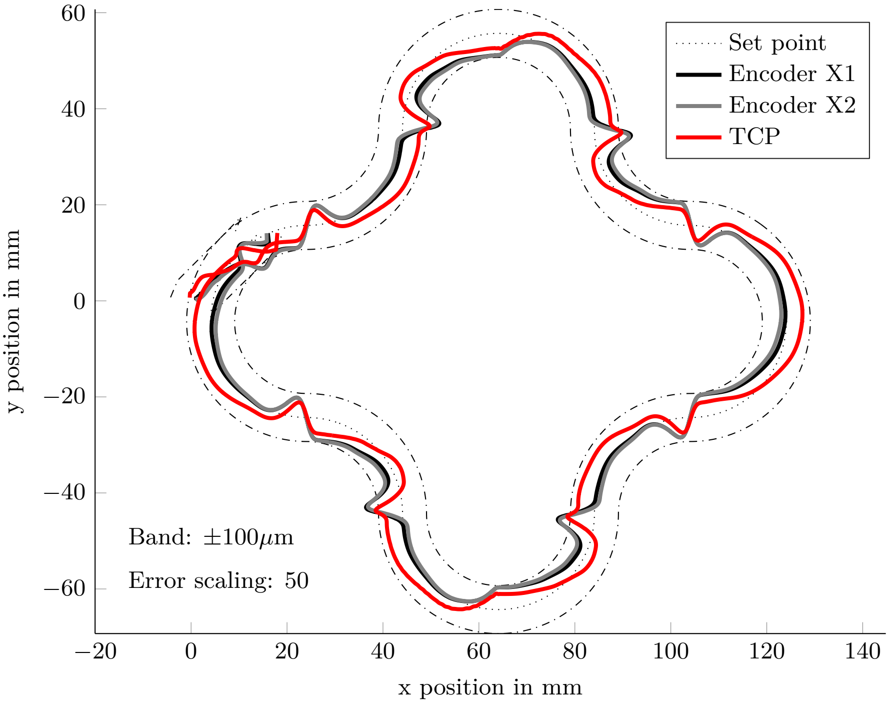
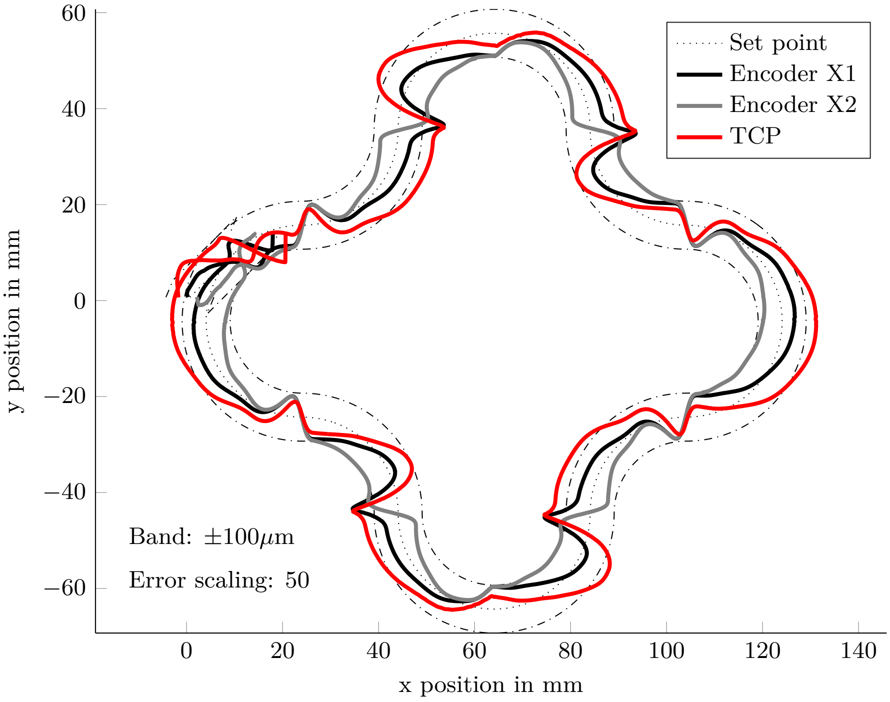


Figure 8.11: Maximum dynamic errors for the X- and Y-axes depending on the position of the Y-axis



(a) $y = 0$



(b) $y = 1300\,\mathrm{mm}$

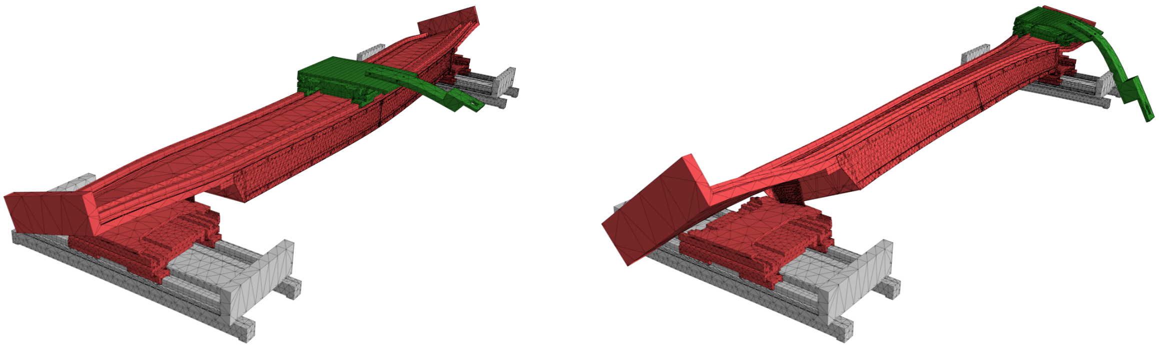
Figure 8.12: Simulated contour errors for a starting position of $x = 0$ and two different Y-axis starting positions

the right end of the travel range. For this position, i.e. for the Y-axis at 1300 mm, the rounded cross contour from *Fig.* 8.7(b) is simulated again with the same controller and friction settings and shown in *Fig.* 8.12(b). For convenience, the contour error for the centre position is shown again in *Fig.* 8.12(a).

There is a significant difference in the accuracy between the contours for both locations of the Y-axis. Moreover, the X1- and X2-axes move almost equally for the centre position but not for the outer position.

For this analysis, the controller parametrisation has not been adapted for different positions, but for the present case, an adaptive control rule could potentially be advantageous and could be studied with the help of the model.

In order to recognise improvement potential for machine structures as well as to gain profound knowledge of the dynamic behaviour of a system, it is important to visualise deformations from simulation results. As an example, *Fig.* 8.13 shows the oscillation shapes for the structure of the test bench under harmonic excitation at the motors of the X-axis at two positions of the Y-axis. In this example, the oscillation shapes for the second resonance frequency are shown. The example shows how an oscillation shape can change within the work-space and that, due to modelling of moving interfaces, the coupling of the moving bodies is rendered correctly.



(a) Harmonic oscillation at $\omega = 752\,\mathrm{rad/s}$ for $y = 0$ (b) Harmonic oscillation at $\omega = 703\,\mathrm{rad/s}$ for $y = 1300\,\mathrm{mm}$

Figure 8.13: Oscillation shapes for harmonic excitation with a total force of 1 N at the motors of the X-axis at the frequency of the second resonance for different Y-axis positions. The deformations are scaled by a factor of 200 000.

8.5 Discussion

In this chapter, a MORe model of a real system has been presented and the quality of the approximation has been assessed. The model approximation for the Andromeda stage has been shown to be satisfactory. Measurement and simulation results in both time and frequency domain are in good accordance. An important fact is that the model has been created without any parameter fitting. This demonstrates that a MORe model can deliver accurate results during the design phase of a new machine without the need for measurements by means of a prototype. This is important for the presented procedure Design to Specifications.

The derivation of requirements used with Design to Specifications has been evaluated for the present example and adjudged to be valuable. The error estimation by means of error transfer function weighting with a MORe model has been examined and approved.

Furthermore, an application of a MORe model for spatial dynamic analysis has been presented which emphasizes the issues with a position-dependent dynamic behaviour of a structure. The analysis by means of a MORe model is very efficient due to the availability of moving interfaces.

9 Conclusion and Outlook

The aim of this dissertation was to develop a comprehensive framework for simulation of the dynamics of machine tools that is on the one hand accurate and on the other hand efficient in terms of both simulation time as well as modelling effort.

A thorough literature study revealed that there were no methods available that enable an efficient model reduction and, even more critical, an efficient error estimation for reduced-order models. Moreover, there were no methods for modelling of moving contacts which are compatible with state-of-the-art model reduction techniques and, furthermore, which allow the application of both forces and torques. It thus follows that there were neither any software tools nor a complete strategy or methods for the efficient set-up of accurate models for simulation of machine tools.

In this thesis, a model reduction method based on projection into combined Krylov and modal subspaces has been presented. Krylov subspaces ensure that the frequency response functions of the reduced system match the ones of the original system at specific frequencies. Modal subspaces ensure that the modes of a system, and thus, the poles of the frequency response function match accurately. Due to beneficial properties of the relative error of the transfer functions of systems reduced by KMS projection, an estimation for the error bound could be developed. A special case of the KMS reduction is equivalent to the widely spread component mode synthesis method. Therefore, the established error estimation can also be used for component mode synthesis.

The KMS based reduction method has been shown to be applicable to systems of first-order structure like thermal systems too. This, however, should be investigated in detail for application of the methods presented here to thermal systems for the analysis of temperature fields and deformations caused by thermal expansion in machine tools.

Modelling of moving interfaces in combination with reduced models is crucial in order to be able to model position dependencies of axes without the need of repeated reduction of the systems for each position. Because, for direct matching of transfer functions, each input and output to a system enhances the order of the reduced system, the number of interfaces is critical, and thus, direct use of finite element nodal DOFs as independent interfaces is not feasible. A new strategy for modelling of moving forces and torques as well as evaluating displacements and rotations at different positions on a flexible body has been presented. The approximation of a trapezoidal function by means of trigonometric interpolation allows to reduce the total number of required inputs and outputs to a minimum. However, because the number of inputs and outputs for moving interfaces is still considerably higher than for stationary interfaces, this method is only practical in combination with a model order reduction method that introduces as few more states per input as possible. The KMS based reduction can be configured such that each input augments the system order exactly by one, what is the minimum for matching the static compliance. The dynamic compliance can, however, still be matched accurately due to the introduction of modes.

In order to additionally increase simulation efficiency, reduced models could possibly be further reduced for specific operation points. Due to superposition of the harmonic terms for moving interfaces at a specific position, the number of interfaces could be decreased considerably and thus also the order of the reduced systems. Even the final assembled composition models consisting of coupled reduced system matrices of the individual bodies could possibly be further reduced before or during transient simulation. Because a composition usually has only a low number of external interfaces, e.g. motors, encoders, and TCP, such a multi-level reduction could significantly reduce the system's order without loss of quality for the results at the interfaces of interest. The reduction could potentially be accomplished in a very short time, because the systems to be reduced are already of comparatively low order. Eventually, the resulting systems could be small enough to enable real-time simulation.

A linearised floating frame of reference formulation for the synthesis of system matrices for flexible bodies in any orientation in space has been presented. This formulation was restricted to the derivation of linearised equations of motion. In order to enable transient simulation including large rotations of the bodies, the non-linear equations of motion for a flexible multi-body system are still to be formulated and implemented.

A comprehensive framework for the simulation of machine tools has been established which allows an effective work-flow. Therefore, the software tool MORe has been created which embeds the newly developed methods together with all necessary functionalities for

- interfacing with the finite element software ANSYS Mechanical,
- definition of stationary and moving interfaces,
- model order reduction,
- composition of multiple flexible bodies in any orientation,
- definition of link properties for standard coupling elements used in mechanical engineering,
- simulative analysis of the structure,
- interfacing to MATLAB and Simulink for simulative analysis of the complete mechatronics system, and
- animation of deformation results.

The amount of time spent for simulations is to a large extent defined by the methods of application of the simulation tools. With the presented Design to Specifications approach, design engineers are not only supplied with a method for distinct and straightforward dynamic performance evaluation but also with a procedure for conscious application of simulation during, e.g. the development of a new machine. Beyond that, systematic derivation of requirements on the mechanical structure from specifications on productivity and accuracy has been presented, what allows to further reduce the number of design iterations.

The presented method for efficient analysis of the dynamic performance is based on weighted error transfer functions, what is restricted to the evaluation of dynamic contour accuracy for feed axes without disturbance forces. For a comprehensive analysis of a machine tool, it would be beneficial to have an integrated tool for automated analysis of

- maximum dynamic contour errors,
- backlash, caused, e.g. by static friction or mechanical play,
- deformation due to static forces,
- deformation due to gravitation, and eventually
- process stability.

This would allow the standardised performance specification for machine tools.

The application of the presented methods and software to a test bench proved the effectiveness and usability of the framework.

The models created with MORe open up a new field of applications which could be researched like, e.g. the calculation of the position-dependent static deformation of high-precision machine tool axes, allowing the calculation of an optimal compensation by scraping or milling profiles, or the analysis of advanced control and trajectory planning algorithms.

Bibliography

[1] J. I. Aliaga, D. L. Boley, R. W. Freund, V. Hernández, 2000, A Lanczos-type Method for Multiple Starting Vectors, Math. Comput., 69, pp. 1577–1601.

[2] Y. Altintas, C. Brecher, M. Weck, S. Witt, 2005, Virtual Machine Tool, CIRP Annals - Manufacturing Technology, 54, pp. 115–138.

[3] D. Amsallem, C. Farhat, 2011, An Online Method for Interpolating Linear Parametric Reduced-Order Models, SIAM Journal on Scientific Computing, 33, pp. 2169–2198.

[4] 2017, ANSYS Mechanical, Release 17. ANSYS, Inc.

[5] A. C. Antoulas, 2005, Approximation of large-scale dynamical systems, SIAM. doi:`https://doi.org/10.1137/1.9780898718713`.

[6] W. E. Arnoldi, 1951, The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem, Quarterly of Applied Mathematics, 9, pp. 17–29.

[7] Z. Bai, K. Meerbergen, Y. Su, 2005, Arnoldi Methods for Structure-Preserving Dimension Reduction of Second-Order Dynamical Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 173–189. doi:`10.1007/3-540-27909-1_7`.

[8] Z. Bai, 2002, Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems, Applied Numerical Mathematics, 43, pp. 9–44.

[9] K.-J. Bathe, J. Dong, 2014, Component mode synthesis with subspace iterations for controlled accuracy of frequency and mode shape solutions, Computers & Structures, 139, pp. 28–32.

[10] K.-J. Bathe, 2002, Finite-elemente-methoden, volume 2, Springer Berlin.

[11] C. Batlle, N. Roqueiro, 2016, Balanced model order reduction for systems depending on a parameter, CoRR, abs/1604.08086.

[12] O. A. Bauchau, 2010, Flexible multibody dynamics, volume 176, Springer.

[13] T. Baudisch, 2003, Simulationsumgebung zur Auslegung der Bewegungsdynamik des mechatronischen Systems Werkzeugmaschine, Ph.D. thesis, Technische Universität München.

[14] T. Bechtold, E. B. Rudnyi, J. G. Korvink, 2004, Error estimation for Arnoldi-based model order reduction of MEMS, System, 10, p. 15.

[15] P. Benner, V. Mehrmann, D. C. Sorensen, 2006, Dimension Reduction of Large-Scale Systems: Proceedings of a Workshop held in Oberwolfach, Germany, 2003: 45 (Lecture Notes in Computational Science and Engineering), Springer.

[16] P. Benner, J. Saak, T. Bonin, M. Zäh, A. Soppa, H. Faßbender, 2011, Moderne Modellordnungsreduktionsverfahren für Finite-Elemente-Modelle zur Simulation von Werkzeugmaschinen, in: Tagungsband Mechatronik 2011.

[17] P. Benner, S. Gugercin, K. Willcox, 2015, A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems, SIAM Review, 57, pp. 483–531.

[18] J. Berkemer, 2003, Gekoppelte Simulation von Maschinendynamik und Antriebsregelung unter Verwendung linearer Finite Elemente Modelle, Ph.D. thesis, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart.

[19] B. N. Bond, 2010, Stability-preserving model reduction for linear and nonlinear systems arising in analog circuit applications, Ph.D. thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science.

[20] T. Bonin, H. Faßbender, A. Soppa, M. Zaeh, 2016, A fully adaptive rational global Arnoldi method for the model-order reduction of second-order MIMO systems with proportional damping, Mathematics and Computers in Simulation, 122, pp. 1–19.

[21] S.-H. Boo, J.-G. Kim, P.-S. Lee, 2016, A simplified error estimator for the {CB} method and its application to error control, Computers & Structures, 164, pp. 53–62.

[22] N. Ceresa, 2016, Messen am TCP, Master's Thesis, ETH Zurich. Supervised by Natanael Lanz.

[23] Y. Chahlaoui, 2010, Two efficient SVD/Krylov algorithms for model order reduction of large scale systems, The University of Manchester, MIMS EPrint: 2010.11, pp. 1–33.

[24] G. R. Cowper, 1973, Gaussian quadrature formulas for triangles, International Journal for Numerical Methods in Engineering, 7, pp. 405–408.

[25] R. R. Craig, M. C. C. Bampton, 1968, Coupling of substructures for dynamic analyses., AIAA Journal, 6, pp. 1313–1319.

[26] R. R. Craig, A. L. Hale, 1988, Block-Krylov component synthesis method for structural model reduction, Journal of Guidance, Control, and Dynamics, 11, pp. 562–570.

[27] R. Craig, Jr., 2000, Coupling of substructures for dynamic analyses: An overview, in: 41st Structures, Structural Dynamics, and Materials Conference and Exhibit, p. 1573.

[28] C. W. De Silva, 1999, Vibration: fundamentals and practice, CRC press.

[29] S. Dietz, K. Knothe, 1997, Reduktion der Anzahl der Freiheitsgrade in Finite-Element-Substrukturen, ILR, Techn. Univ.

[30] S. Dietz, O. Wallrapp, S. Wiedemann, 2003, Nodal vs. modal representation in flexible multibody system dynamics, Proceedings of multibody dynamics, , pp. 1–4.

[31] V. Druskin, V. Simoncini, M. Zaslavsky, 2014, Adaptive Tangential Interpolation in Rational Krylov Subspaces for MIMO Dynamical Systems, SIAM Journal on Matrix Analysis and Applications, 35, pp. 476–498.

[32] R. Eid, B. Salimbahrami, B. Lohmann, E. B. Rudnyi, J. G. Korvink, 2006, Parametric Order Reduction of Proportionally Damped Second Order Systems, Technical Reports on Automatic Control, TRAC-1, pp. 1–11.

[33] R. Eid, 2009, Time domain model reduction by moment matching, Ph.D. thesis, Universität München.

[34] K. Elssel, H. Voss, 2006, An A Priori Bound for Automated Multilevel Substructuring, SIAM Journal on Matrix Analysis and Applications, 28, pp. 386–397.

[35] J. Fehr, 2011, Automated and Error Controlled Model Reduction in Elastic Multibody Systems, Ph.D. thesis, Institut für Technische und Numerische Mechanik, Universität Stuttgart.

[36] C. A. Felippa, 2004, Introduction to finite element methods, University of Colorado.

[37] A. Fischer, P. Eberhard, 2011, Simulation-based stability analysis of a thin-walled cylinder during turning with improvements using an adaptronic turning chisel, Archive of Mechanical Engineering, 58, pp. 367–391.

[38] M. Fischer, P. Eberhard, 2013, Application of parametric model reduction with matrix interpolation for simulation of moving loads in elastic multibody systems, Advances in Computational Mathematics, , pp. 1–24.

[39] M. Fischer, P. Eberhard, 2014, Linear model reduction of large scale industrial models in elastic multibody dynamics, Multibody System Dynamics, 31, pp. 27–46.

[40] M. Fischer, P. Eberhard, 2014, Simulation of Moving Loads in Elastic Multibody Systems With Parametric Model Reduction Techniques, Archive of Mechanical Engineering, 61, pp. 209–226.

[41] M. Fischer, A. Vasilyev, T. Stykel, P. Eberhard, Model Order Reduction for Elastic Multibody Systems with Moving Loads.

[42] G. Flagg, C. Beattie, S. Gugercin, 2012, Convergence of the Iterative Rational Krylov Algorithm, Systems & Control Letters, 61, pp. 688–691.

[43] R. W. Freund, 2000, Krylov-subspace methods for reduced-order modeling in circuit simulation, Journal of Computational and Applied Mathematics, 123, pp. 395–421. Numerical Analysis 2000. Vol. III: Linear Algebra.

[44] K. Gallivan, A. Vandendorpe, P. V. Dooren, 2004, Model Reduction of MIMO Systems via

Tangential Interpolation, SIAM Journal on Matrix Analysis and Applications, 26, pp. 328–349.

[45] I. M. Garcia-Herreros Landazabal, 2012, Méthode de Modélisation et de Commande des Systèmes de Positionnement Multi-actionnés de type Axe en Gantry, Ph.D. thesis, École Nationale Supérieure d'Arts et Métiers.

[46] E. J. Grimme, 1997, Krylov projection methods for model reduction, Ph.D. thesis, University of Illinois at Urbana-Champaign.

[47] S. Gugercin, C. Beattie, A. C. Antoulas, 2006, A Rational Krylov Iteration for Optimal H_2 Model Reduction, in: Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems.

[48] R. J. Guyan, 1965, Reduction of stiffness and mass matrices, AIAA Journal, 3, p. 380.

[49] G. H. K. Heirman, T. Tamarozzi, W. Desmet, 2011, Static modes switching for more efficient flexible multibody simulation, International Journal for Numerical Methods in Engineering, 87, pp. 1025–1045.

[50] W. C. Hurty, 1965, Dynamic Analysis of Structural Systems Using Component Modes, AIAA Journal, 3, pp. 678–685.

[51] H. Jakobsson, M. G. Larson, 2012, A Posteriori Error Analysis of Component Mode Synthesis for the Frequency Response Problem, arXiv preprint arXiv:1206.5479,

[52] E. Jones, T. Oliphant, P. Peterson, 2001–, SciPy: Open source scientific tools for Python. [Online; accessed 06.09.2017].

[53] A. A. Kadir, X. Xu, E. Hämmerle, 2011, Virtual machine tools and virtual machiningA technological review, Robotics and Computer-Integrated Manufacturing, 27, pp. 494–508.

[54] J.-G. Kim, P.-S. Lee, 2014, An accurate error estimator for Guyan reduction, Computer Methods in Applied Mechanics and Engineering, 278, pp. 1 – 19.

[55] J.-G. Kim, P.-S. Lee, 2015, Posteriori Error Estimation Method for Flexibility-Based Component Mode Synthesis, AIAA Journal, 53, pp. 2828–2837.

[56] K. Knothe, H. Wessels, 1999, Finite Elemente: Eine Einführung für Ingenieure, Springer Berlin Heidelberg, Berlin, Heidelberg. doi:`10.1007/978-3-662-07235-6_1`.

[57] D. Kono, S. Weikert, A. Matsubara, K. Yamazaki, 2012, Estimation of dynamic mechanical error for evaluation of machine tool structures, Int. J. of Automation Technology, 6, pp. 147–153.

[58] D. Kubalinska, 2008, Optimal interpolation-based model reduction, Ph.D. thesis, Universität Bremen.

[59] P. Lancaster, I. Zaballa, 2009, Diagonalizable quadratic eigenvalue problems, Mechanical Systems and Signal Processing, 23, pp. 1134–1144.

[60] C. Lanczos, 1950, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, Journal of Research of the National Bureau of Standards, 45, p. 255.

[61] N. Lang, J. Saak, P. Benner, 2014, Model Order Reduction for Systems with Moving Loads, at Automatisierungstechnik, 62, pp. 512–522. 7.

[62] N. Lanz, D. Spescha, A. Ryser, N. Ceresa, S. Weikert, 2017, Modelling and Validation of Position Dependant Structural Deformations of a Machine Tool Structure under Gravitational Loads, euspen's 17th International Conference and Exhibition, Hannover,

[63] M. Lehner, 2007, Modellreduktion in elastischen Mehrkörpersystemen, Ph.D. thesis, Institut für Technische und Numerische Mechanik, Universität Stuttgart.

[64] R. B. Lehoucq, D. C. Sorensen, C. Yang, ARPACK: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods, Available from netlib@ornl.gov, 1997.

[65] R. B. Lehoucq, D. C. Sorensen, C. Yang, 1998, ARPACK users' guide : solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods, Philadelphia : SIAM.

[66] X. S. Li, 2005, An Overview of SuperLU: Algorithms, Implementation, and User Interface, ACM Transactions on Mathematical Software, 31, pp. 302–325.

[67] X. S. Li, 2005, An Overview of SuperLU: Algorithms, Implementation, and User Interface, ACM Trans. Math. Softw., 31, pp. 302–325.

[68] B. Lohmann, B. Salimbahrami, 2004, Structure Preserving Reduction of Large Second Order Models by Moment Matching, PAMM, 4, pp. 572–573.

[69] T. Lorenzer, S. Weikert, K. Wegener, 2007, Decision-making aid for the design of reconfigurable machine tools, in: 2nd International Conference on Changeable, Agile, Reconfigurable and Virtual Production, Toronto, pp. 720–729.

[70] T. Lorenzer, 2011, Wandelbarkeit in der Serienfertigung durch rekonfigurierbare Werkzeugmaschinen, Ph.D. thesis, Institut für Werkzeugmaschinen und Fertigung, Eidgenössische Technische Hochschule ETH Zürich.

[71] P. Maglie, 2011, Parallelization of Design and Simulation: Virtual Machine Tools in Real Product Development, Ph.D. thesis, ETH No. 19801, Institut für Werkzeugmaschinen und Fertigung, Eidgenössische Technische Hochschule ETH Zürich. doi:`https://doi.org/10.3929/ethz-a-000093833`.

[72] 2016, MATLAB, Release 2016b. The MathWorks.

[73] 2015, ME'scope, Version 6.0. Vibrant Technology, Inc.

[74] L. Myklebust, B. Skallerud, 2002, Model Reduction Methods for Fexible Structures, Nordic Seminar on Computational Mechanics, 15, pp. 171–175.

[75] W. Neithardt, 2004, Methodik zur Simulation und Optimierung von Werkzeugmaschinen in der Konzept- und Entwurfsphase auf Basis der Mehrkörpersimulation, Ph.D. thesis, Universität Karlsruhe (TH).

[76] C. Nowakowski, J. Fehr, M. Fischer, P. Eberhard, 2012, Model order reduction in elastic multibody systems using the floating frame of reference formulation, IFAC Proceedings Volumes, 45, pp. 40–48.

[77] T. Oertli, 2008, Strukturmechanische Berechnung und Regelungssimulation von Werkzeugmaschinen mit elektromechanischen Vorschubantrieben, Ph.D. thesis, Technische Universität München.

[78] H. Panzer, J. Mohring, R. Eid, B. Lohmann, 2010, Parametric Model Order Reduction by Matrix Interpolation, at - Automatisierungstechnik, 58.

[79] K. C. Park, Y. H. Park, 2004, Partitioned Component Mode Synthesis via a Flexibility Approach, AIAA Journal, 42, pp. 1236–1245.

[80] R. Pinnau, 2008, Model Reduction via Proper Orthogonal Decomposition, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 95–109. doi:`10.1007/978-3-540-78841-6_5`.

[81] 2017, Python Language Reference, Version 2.7. Python Software Foundation.

[82] P. Ramachandran, G. Varoquaux, 2011, Mayavi: 3D Visualization of Scientific Data, Computing in Science & Engineering, 13, pp. 40–51.

[83] S. S. Rao, 2005, The Finite Element Method In Engineering, Butterworth-Heinemann.

[84] J. W. S. B. Rayleigh, 1877, The Theory of Sound, MacMillan and Company.

[85] S. Rubin, 1975, Improved Component-Mode Representation for Structural Dynamic Analysis, AIAA Journal, 13, pp. 995–1006.

[86] E. B. Rudnyi, J. G. Korvink, 2006, Model Order Reduction for Large Scale Engineering Models Developed in ANSYS, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 349–356. doi:`10.1007/11558958_41`.

[87] A. Ryser, 2016, Modifikation und Modellierung eines Dynamikprüfstands, Bachelor's Thesis, ETH Zurich. Supervised by Natanael Lanz.

[88] B. Salimbahrami, B. Lohmann, 2002, Krylov subspace methods in linear model order reduction: introduction and invariance properties, in: Sci. Rep.. Inst. of Automation, Citeseer.

[89] B. Salimbahrami, B. Lohmann, 2005, Second order Krylov subspace for the reduction of second order systems, in: Methods and Applications in Automation, 25th-26th Colloquium of Automation, Salzhausen, Germany, No. 1.1 in Publication Series of the Institute of Automation, University of Bremen, Shaker Verlag Aachen, pp. 1–11.

[90] B. Salimbahrami, B. Lohmann, T. Bechtold, 2002, Two-sided arnoldi in order reduction of large scale MIMO systems, Sci. Report, IAT, Univ. Bremen,

[91] B. Salimbahrami, R. Eid, B. Lohmann, 2006, Model reduction by second order Krylov subspaces: Extensions, stability and proportional damping, in: 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, IEEE, pp. 2997–3002.

[92] W. Schiehlen, P. Eberhard, 2014, Applied Dynamics, Springer.

[93] W. H. Schilders, H. A. Van der Vorst, J. Rommes, 2008, Model order reduction: theory, research aspects and applications, volume 13, Springer.

[94] L. Schumaker, 2007, Spline Functions: Basic Theory, Cambridge Mathematical Library, 3 ed., Cambridge University Press. doi:`10.1017/CBO9780511618994`.

[95] R. Schwertassek, O. Wallrapp, 1999, Dynamik flexibler Mehrkörpersysteme, Springer Vieweg.

[96] A. A. Shabana, R. Schwertassek, 1998, Equivalence of the floating frame of reference approach and finite element formulations, International Journal of Non-Linear Mechanics, 33, pp. 417–432.

[97] A. A. Shabana, 1997, Flexible Multibody Dynamics: Review of Past and Recent Developments, Multibody System Dynamics, 1, pp. 189–222.

[98] A. A. Shabana, 2005, Dynamics of Multibody Systems, 3 ed., Cambridge University Press.

[99] D. Siedl, 2008, Simulation des dynamischen Verhaltens von Werkzeugmaschinen während Verfahrbewegungen, Ph.D. thesis, Institut für Werkzeugmaschinen und Fertigung, Technische Universität München.

[100] 2016, Simulink, Release 2016b. The MathWorks.

[101] B. Simeon, 2006, On Lagrange multipliers in flexible multibody dynamics, Computer Methods in Applied Mechanics and Engineering, 195, pp. 6993 – 7005. Multibody Dynamics Analysis.

[102] J. T. Spanos, M. H. Milman, D. L. Mingori, 1992, A new algorithm for L2 optimal model reduction, Automatica, 28, pp. 897–909.

[103] D. Spescha, S. Weikert, O. Zirn, K. Wegener, 2017, Synchronisation of Feed Axes with Differing Bandwidths Using Set Point Delay, International Journal of Automation Technology, 11, pp. 155–164.

[104] P. Steinke, 2010, Finite-Elemente-Methode, 3. ed., Springer.

[105] T. Stykel, A. Vasilyev, 2016, A two-step model reduction approach for mechanical systems with moving loads, Journal of Computational and Applied Mathematics, 297, pp. 85–97.

[106] Q. Su, V. Balakrishnan, C.-K. Koh, 2002, Efficient approximate balanced truncation of general large-scale RLC systems via Krylov methods, in: Proceedings of the 2002 Asia and South Pacific Design Automation Conference, IEEE Computer Society, p. 311.

[107] T. Tamarozzi, P. Ziegler, P. Eberhard, W. Desmet, 2013, Static modes switching in gear contact simulation, Mechanism and Machine Theory, 63, pp. 89–106.

[108] 2017, Traits, Version 4. Enthought, Inc.

[109] 2017, TraitsUI, Version 4. Enthought, Inc.

[110] M. J. Turner, R. W. Clough, H. C. Martin, L. J. Topp, 1956, Stiffness and deflection analysis of complex structures, Journal of the Aeronautical Sciences, 23, pp. 805–823.

[111] P. Van Dooren, K. A. Gallivan, P.-A. Absil, 2008, H2-optimal approximation of MIMO linear dynamical systems, arXiv preprint arXiv:0807.4807,

[112] M. Wabner, U. Frieß, S. Hofmann, A. Hellmich, J. Quellmalz, 2014, Optimierte Inbetriebnahme durch Simulation, wt Werkstattstechnik online, 1/2, pp. 97–99.

[113] S. Weikert, C. Jaeger, S. Bossoni, K. Wegener, 2008, Efficient evaluation of early machine concepts in the frequency domain including control issues, in: ASPE Annual Meeting.

[114] S. Weikert, 2000, Beitrag zur Analyse des dynamischen Verhaltens von Werkzeugmaschinen, Ph.D. thesis, ETH Nr. 13596, ETH Zürich. doi:`https://doi.org/10.3929/ethz-a-003896403`.

[115] W. Witteveen, 2012, Comparison of CMS, Krylov and balanced truncation based model reduction from a mechanical application engineer's perspective, in: Topics in Experimental Dynamics Substructuring and Wind Turbine Dynamics, Volume 2, Springer, pp. 319–331.

[116] T. Wolf, H. K. F. Panzer, B. Lohmann, 2013, H2 Pseudo-Optimality in Model Order Reduction by Krylov Subspace Methods, in: 2013 European Control Conference (ECC), pp. 3427–3432.

[117] Y. Xu, T. Zeng, 2011, Optimal H2 Model Reduction for Large Scale MIMO Systems via Tangential Interpolation, International Journal of Numerical Analysis and Modeling, 8, pp. 174–188.

[118] M. Zaeh, T. Oertli, J. Milberg, 2004, Finite Element Modelling of Ball Screw Feed Drive Systems, CIRP Annals - Manufacturing Technology, 53, pp. 289 – 292.

[119] O. Zirn, 2008, Machine Tool Analysis – Modelling, Simulation and Control of Machine Tool Manipulators, Habilitation thesis, Department of Mechanical and Process Engineering, ETH Zürich. doi:`10.3929/ethz-a-005825192`.

List of Publications

(a) D. Spescha, S. Weikert, S. Retka, K. Wegener, Krylov and Modal Subspace based Model Order Reduction with A-Priori Error Estimation, preprint, https://doi.org/10.3929/ethz-b-000284435

(b) D. Spescha, S. Weikert, K. Wegener, Modelling of Moving Interfaces for Reduced-Order Finite Element Models using Trigonometric Interpolation, preprint, https://doi.org/10.3929/ethz-b-000284509

(c) D. Spescha, S. Weikert, K. Wegener, Simulation of the dynamic behaviour of machine tools, 16th Mechatronics Forum International Conference, 2018, in press

(d) D. Spescha, S. Weikert, K. Wegener, Design to Specifications - A Strategy for Specification-Based Machine Design, 8th CIRP Conference on High Performance Cutting (HPC 2018), 2018, Budapest

(e) D. Spescha, S. Weikert, O. Zirn, K. Wegener, Synchronisation of Feed Axes with Differing Bandwidths Using Set Point Delay, International Journal of Automation Technology, 2017, 11, 155–164, https://doi.org/10.20965/ijat.2017.p0155

(f) N. Lanz, D. Spescha, S. Weikert, K. Wegener, Efficient Static and Dynamic Modelling of Machine Structures with Large Linear Motions, International Journal of Automation Technology, 2018, 12, in press

(g) M. Kuffa, N. Lanz, D. Spescha, K. Wegener, Efficient dynamic machine structure modelling for high performance dry grinding, 8th CIRP Conference on High Performance Cutting (HPC 2018), 2018, Budapest

(h) N. Lanz, D. Spescha, A. Ryser, N. Ceresa, S. Weikert, Modelling and Validation of Position Dependant Structural Deformations of a Machine Tool Structure under Gravitational Loads, euspen's 17th International Conference and Exhibition, 2017, Hannover

(i) J. Mayr, M. Ess, F. Pavliek, S. Weikert, D. Spescha, W. Knapp, Simulation and measurement of environmental influences on machines in frequency domain, CIRP Annals - Manufacturing Technology, 2015, Kapstadt, 479–482

(j) D. Spescha, H. Nguyen, Mechatronische Mehrkörpersimulation - Ansatz zur Behandlung wandernder Koppelstellen im FEM, VPE Swiss Workshop - Kopplung von Struktur und Steuerung/Regelung, 2014, Rapperswil, https://doi.org/10.3929/ethz-a-010185439

(k) D. Spescha, R. Altenburger, R. Thiel, Entwicklung und Analyse eines Dynamometers in Hexapod-Struktur, Internationales Forum Mechatronik, 2010, Winterthur

Curriculum Vitae

Daniel Spescha

Date of Birth: 28 April 1984

Place of Birth: Ilanz/Glion, Switzerland

Citizenship: Switzerland

Education

2008 – 2011	Master of Science ZFH in Engineering (part-time 50%) Zurich University of Applied Sciences ZHAW, Winterthur, Switzerland
2005 – 2008	Degree program in electrical engineering, specialisation in mechatronics Zurich University of Applied Sciences ZHAW, Winterthur, Switzerland
2000 – 2004	Apprenticeship as automation technician Kraftwerke Vorderrhein AG, Tavanasa, Switzerland
2000 – 2004	Professional maturity certificate Gewerbliche Berufsschule, Chur, Switzerland
1997 – 2000	Secondary school Rueun, Switzerland
1991 – 1997	Primary school Rueun, Switzerland

Employment History

since 10.2011	Research associate Inspire AG, Corporation for Mechatronic Systems and Manufacturing Technology, Swiss Federal Institute of Technology ETH, Zurich, Switzerland
06.2011 – 09.2011	Research associate Institute of Mechatronic Systems IMS, Zurich University of Applied Sciences ZHAW, Winterthur, Switzerland
2008 – 2010	Research assistant (part-time 50%) Institute of Mechatronic Systems IMS, Zurich University of Applied Sciences ZHAW, Winterthur, Switzerland
2004 – 2005	Automation technician Hydro Surselva AG, Ilanz/Glion, Switzerland